AFIT/GAM/ENC/95D-1

Adaptive and Fixed Wavelet Features for Narrowband Signal Classification

THESIS

Antony J. Pohl
First Lieutenant, USAF

AFIT/GAM/ENC/95D-1

AFIT/GAM/ENC/95D-1

Adaptive and Fixed Wavelet Features for Narrowband Signal Classification

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Applied Mathematics

Antony J. Pohl, B.A.
First Lieutenant, USAF

December 1995

Acknowledgements

First I would like thank my wife Jennifer for the love and patience she exhibited during my studies at AFIT. Without her support I could not have put forth my best effort on this work.

Next I would like to thank my thesis committee members Maj Dennis Ruck, Ph.D., and Dr. Mark Oxley for their guidance and for always being there to answer my "quick questions." I would especially like to thank my advisor, Maj Gregory Warhola, Ph.D., for his exceptional inspiration and guidance which pushed me beyond where I would have otherwise taken this work.

Finally, I would like to thank the members of the pattern recognition corner in the signal processing laboratory for their companionship over the past six months.

Antony J. Pohl

Table of Contents

List of Figures

List of Tables

Abstract

The application of the multiresolution analysis developed by Mallat to signal classification by Pati and Krishnaprasad and Szu, *et al*, is further explored in this thesis. Several different wavelet-based feature extraction and classification systems are developed and implemented. Methods which rely on the traditional dyadic wavelet decomposition and on the adaptive wavelet representation are presented. Each of the classification systems is implemented for a labeled data set of narrowband signals. Finally, classification results on the full data set and on low frequency Fourier coefficients are provided as baseline comparisons for our work.

Adaptive and Fixed Wavelet Features for Narrowband Signal Classification

I. Introduction

Artificial Neural Networks (ANN) have shown success in solving classification problems. However, in designing a classification system there are several choices that needed to made. First, a decision needs to be made on the particular neural network model and training method. Then, a particular set of features are extracted using a particular extraction method. Finally, a choice is made on method of validation which gives some bound on the classification error rate. Unfortunately, there exist only loose guidelines which govern any of these choices [1] [2]. Thus, decisions are often made which influence the classification success percentage of the classifier based on little more than intuition or even random chance.

Recently, the theory of wavelets has emerged as an alternate time-frequency analysis tool to the Fourier transform. Wavelets have been applied to a variety of problems, most notably data compression and noise reduction. Hence, it is reasonable to investigate the application of the theory of wavelets to the problem of feature extraction.

1.1 Background

In researching this thesis the goal was to build a classification system for pulsed narrowband signals; i.e., signals with slowly varying amplitude and phase. In particular, we would like to be able to label and extract a single pulse from a stream of time samples and classify it as being of a specific class. Since the extracted pulse may itself consist of many time samples, it may not be feasible to work with the full set of data. it was therefore decided to concentrate on the feature extraction process. As it is discussed in Chapter II, Pati and Krishnaprasad [3] and Szu, *et al*, [4] offer two different approaches for representing signals in terms of a

wavelet functions. The multilayer classification example in Kadambe and Srinivasan's article [5] was used as a foundation from which to investigate the goal of a classification system for narrowband signals.

1.2 Objective

Demonstrate the ability of a wavelet-based feature extraction and classification system to classify narrowband signals using both adaptive and fixed wavelets.

1.3 Approach/Methodology

A wavelet-based feature extraction and classification system will be developed for narrowband signals with a high ratio of data samples to features. Once the wavelet based feature extraction and classification system is developed, it use will be demonstrated by comparing it to the classification rate achievable by classification on all of the original data and on features extracted with Fourier methods from the original data.

1.4 Equipment and Materials

This thesis requires no special materials or equipment. SPARC 5 and SPARC 20 workstations are used to support all programming. More specifically, LaTeX is used to typeset this document. Matlab is used for generating plots and some general purpose programming. LNKnet is used for all multilayer perceptron applications. All general programming that is computationally intense is done in the Kernighan & Ritchie C language.

1.5 Notation

We use the following notation throughout this thesis:

- $\mathbf{C}$ for the set of complex numbers.
- $\mathbf{Z}$ for the set of integers.
- $\mathbf{Z}^+$ for the set of non negative integers.

- $\mathbf{R}$ for the set of real numbers.
- $\mathbf{L}^2(\mathbf{R})$ for the space of measurable, square-integrable functions:

$$\mathbf{L}^2(\mathbf{R}) = \{f : \mathbf{R} \rightarrow \mathbf{C} \mid f \text{ is Lebesgue-measureable and } \int_{-\infty}^{+\infty} |f(x)|^2 dx < \infty\}. \tag{1.1}$$

  If $f \in \mathbf{L}^2(\mathbf{R})$, f is sometimes referred to as a finite-energy function.

- $l^2(\mathbf{Z})$ for the space of square-summable sequences:

$$l^2(\mathbf{Z}) = \left\{ a = (\ldots, a_{-1}, a_0, a_1, \ldots) \ : \ a_k \in \mathbf{C}, \sum_{k=-\infty}^{+\infty} |a_k|^2 < \infty \right\}. \tag{1.2}$$

  For matrices and operators $\mathbf{A}$, we use the following notation:

- $\mathbf{A} = [a(i,j)]$ defines a matrix $\mathbf{A}$ whose element in the i-th row and j-th column is given by $a(i,j)$, where a is a function on $\mathbf{Z}^+ \times \mathbf{Z}^+$.
- $\mathbf{A}^{\mathrm{T}}$ for the transpose of the matrix $\mathbf{A}$.
- $\mathbf{v} = [v(i)]$ defines a column vector $\mathbf{v}$ whose element in the i-th row is given by $v(i)$, where v is a function on $\mathbf{Z}^+$.
- $\sum_n$ will denote the sum over all $n \in \mathbf{Z}$ unless specific limits are given.
- The Fourier transform of f will be denoted by either $\hat{f}$ or F. It is defined as $\hat{f}(\nu) = \int_{-\infty}^{+\infty} f(x) e^{-i2\pi\nu x} dx$ for $f \in \mathbf{L}^2(\mathbf{R})$ and as $\hat{f}_k = \sum_n f_n e^{-i2\pi kn}$ for $f \in l^2(\mathbf{Z})$.

1.6 Scope

This thesis is limited to the following:

1. A brief description of the mathematical theory of wavelets and multiresolution analysis as applied to neural networks and the multilayer perceptron.
2. A development of a wavelet-based feature extraction and classification system.

3. Development of the tools necessary to implement the wavelet-based feature extraction and classification system.
4. An application of the wavelet-based feature extraction and classification system to real world data to demonstrate the performance of the system.

1.7 Overview of Thesis

In Chapter II the current theory which leads to the development of methods to be used in this thesis is reviewed. In Chapter III the methods are examined with regard to their mathematical foundations and provide simple computational examples. A report on experimental classification outcomes is provided in Chapter IV. Conclusions of this work as well as recommendations for future work are discussed in Chapter V.

II. Background Theory

2.1 Introduction

Chapter II contains a description of the various methods used in the field and builds the methods for use in this thesis. It serves as a literature review.

2.2 Wavelet Neural Networks as Function Approximators

2.2.1 Pati and Krishnaprasad. Pati and Krishnaprasad [3] describe a network in which the sigmoidal activation functions of a typical neural network are replaced by particular shifts and dilations of a given mother wavelet. Thus, consider equation 2.1 where $\mathbf{T}$, a closed proper subset of $\mathbf{R} \times \mathbf{R}$, is the set of all training pairs (x, y):

$$y \approx f(x) = \sum_{m,n} w_{m,n}\psi_{m,n}(x)\ ,\ \forall (x,y) \in \mathbf{T},\ w_{m,n}, x, y \in \mathbf{R},\ m \in \mathbf{Z},\ n \in \mathbf{Z}^+, \quad (2.1)$$

where "$\approx$" is defined such that there exists $\epsilon \in \mathbf{R}^+$ so that

$$\epsilon > |f(x) - y|^2, \quad (2.2)$$

and where $\psi_{m,n}$ is a wavelet such that

$$\psi_{mn}(x) = 2^{-m/2}\psi(2^{-m}x - n). \quad (2.3)$$

Pati's network is similar to the general expression of the discrete wavelet transform. We now have a network structure which is simply a projection onto a basis – an inner product – where the basis is a wavelet basis. When we talk about learning a given "training" set, we are really just projecting the training vectors onto the wavelet basis. Since an infinite basis cannot be implemented, a finite subset over the compactly supported interval on which the training data is defined is chosen. Furthermore, we also limit the set to a maximum dilation. Define $\mathbf{I}$ as the finite set of all shifts and dilations (m, n). Then we now can approximate the training

data by the finite set of shifts and dilations $(m, n) \in \mathbf{I}$ and a corresponding set of coefficients (or weights) $\{w_{m,n}\}_{(m,n)\in\mathbf{I}} \subset \mathbf{R}$ [3].

The overall approximation error is determined by

$$E = \sum_{(x,y)\in\mathbf{T}} |f(x) - y|^2 \tag{2.4}$$

This error functional is nearly identical to that of the backpropagation algorithm with only one important difference. It turns out that the error functional described above is convex in terms of the weights $w_{m,n}$. This is quite different from the backpropagation algorithm, which, in general, has a non-linear error surface.

Due to the convexity of the error functional, any minimizer is a global minimizer. Furthermore, it is clear that simple iterative schemes such as gradient descent perform adequately since there is no possibility of getting stuck in local minima. Pati further states that the weights may be obtained by considering the fact that minimizing E as defined above defines a least squares problem. The solution can therefore be determined by solving the system of linear equations constructed by the first order optimality condition $\frac{\partial E}{\partial w_{m,n}} = 0$ at the optimal weight [3].

The authors present two network synthesis algorithms. The first algorithm involves determining the set of wavelets for use as activation functions for the hidden layer neurons by considering the time and frequency limits of the training data. Given that the training data is bounded in both time and frequency, the exact shifts and dilations of the mother wavelet can be determined which are necessary to adequately cover the time and frequency range of the training data. This number is the upper bound of hidden layer neurons necessary to approximate the functional relationship between x and y to any precision ϵ. Unfortunately, this method can be computationally intractable if the number of required wavelets is very large; i.e., the time and frequency bounds are very large. The second synthesis algorithm addresses this problem by starting out at a low dilation and gradually refining the set of wavelets at higher dilation for the regions of the training data that exhibit localized high

frequency behavior. The network coefficients must be learned for the initial dilation. Then additional wavelets (neurons) are added wherever the coefficients exhibit a local minimum. Finally, the network coefficients are learned once again for the augmented set of wavelets. This procedure is repeated until the approximation error is less than ϵ.

Note that the networks considered so far were for one dimensional training sets; i.e., $(x, y) \in \mathbf{I}$ where $x, y \in \mathbf{R}$. Pati states that an extension to higher dimensions, $(\mathbf{x}, y) \in \mathbf{I}$ where $\mathbf{x} \in \mathbf{R}^n, y \in \mathbf{R}$, is straightforward but potentially computationally expensive.

2.2.2 Zhang. In a paper presented at the 32nd Conference on Decision and Control, Zhang [6] describes an implementation of a wavelet neural network based on Pati and Krishnaprasad's [3] first synthesis algorithm and the orthonormal least squares minimization method. Zhang proposes to build a candidate set of wavelets based from the initial infinite set of all possible shifts and dilations of the mother wavelet by first truncating it to a finite set based on some *a priori* knowledge about the training data. The criteria are given by the the time and frequency support of the training data set. The resulting set is a subset of the regular pyramid structure of wavelets usually associated with a dyadic multiresolution decomposition. The goal is to select N wavelets from the candidate set, such that these N are optimal with respect to approximation error [6].

Starting with the network equation

$$z(x) = \sum_{\lambda \in \Lambda} w_\lambda \psi_\lambda(x), \tag{2.5}$$

where $\Lambda \subset \{1, 2, ..., M\}$ is an index set which is used to label the candidate set of wavelets, Zhang derives the criterion that needs to be maximized in order to minimize

$$E = \sum_{(x,y) \in \mathbf{T}} |z(x) - y|^2. \tag{2.6}$$

His method involves using the Gram-Schmidt orthonormalization method to determine the N wavelets and their shift and dilation parameters. Finally, the weights are calculated by a simple inversion of an upper triangular matrix.

2.2.3 Szu, Telfer and Kadambe. In contrast to the networks proposed by Pati and Krishnaprasad and Zhang, Szu, *et al*, [4] do not fix the shift and dilation parameters, initially choosing only a particular mother wavelet. After empirically determining the desired network size, the weights, shifts, and dilations are adaptively calculated. Thus, whereas Szu's mother wavelet may lead to an orthonormal basis using only integer shifts and dilations, it follows that we will, in general, be dealing with a frame [4] [7] [8]. The following equation describes what Szu calls the Adaptive Wavelet Representation (AWR) network:

$$y(t) = \sum_{n=1}^{N} w_n h(\frac{t - b_n}{a_n});\ t = 1, \ldots, T;\ w_n, b_n \in \mathbf{R};\ a_n \in \mathbf{R} - \{0\}, \tag{2.7}$$

where $h \in \mathbf{L}^2(\mathbf{R})$ is a wavelet. Unfortunately though, this method leads to an error surface which is non-linear. It is therefore possible to encounter problems associated with local minima [4] [9].

2.2.4 Kadambe and Srinivasan. Kadambe and Srinivasan [5] take the AWR network developed by Szu, *et al*, and use it in conjunction with a one-layer backpropagation neural network to classify speech signals. Their approach is to first find a close approximation to an input signal in terms of a fixed number of adaptive wavelets, where the approximation represents the projection of the input signal onto the function space spanned by the adaptive wavelets. Then, all parameters – the weights, shifts, and dilations – are fed to a one-layer backpropagation neural network for classification. The interesting feature of this classification system is also its downfall. The fact that each input signal is represented to a minimum squared error by a number of adaptive wavelets whose parameters are used for classification is both novel and promising – as shown in the article. However, the downside is that a non-linear optimization problem, the AWR network, must be solved for each input signal during the

testing phase. This means that this system may take a very long time during testing and will certainly not be implementable in real time on today's computers.

This classification system forms the basis for research from which we develop our classification system.

2.3 Pattern Recognition

Pattern Recognition is a discipline which utilizes a set of features or characteristics measured from the object in order to classify a particular object. For example, at a tuna processing plant it would be prudent to separate the tuna from all other fish that were also caught in the nets. We could choose to examine the color of each fish under the assumption that each tuna would fall within a certain color range most of the time, whereas other fish such as salmon should be similar in color to each other but not to the tuna. The color measurement is be distributed according to a class conditional probability distribution. If the distributions are well separated, discrimination is simple. However, if the distributions overlap as in figure 2.1, then a decision boundary must be set. Depending on the measurement of color and the decision boundary, the classifier labels the fish as one class or the other. The probability of error is related to the class conditional probabilities of being on the wrong side of the decision boundary for a given class.

A typical pattern recognition system is composed of several sections. The typical layout is data gathering, segmentation, feature selection/reduction, and classification [10]. Figure 2.2 shows the typical pattern recognition system.

Segmentation is defined as separating the important data from all the data gathered. In this thesis, after preprocessing the raw data by demodulating it according to the Double Sideband–Suppressed Carrier demodulation algorithm, we segment the data by extracting individual pulses. Features are measured or calculated from the data and then these are used to make a decision on the class of the sample. There are often too many features in the gathered data and hence it is necessary to process the data in order to reduce the number of features to a manageable size. We concentrate our research on feature selection/reduction with

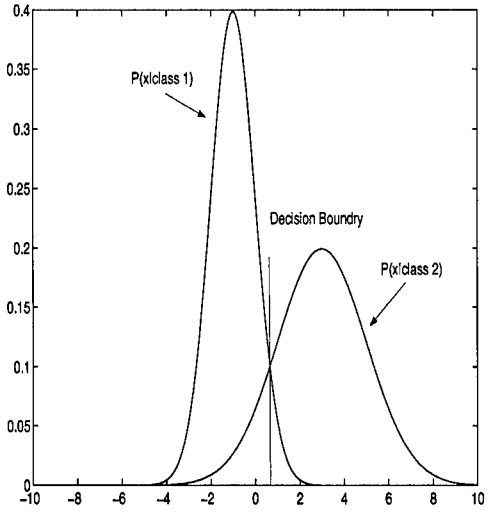

Figure 2.1 Class Conditional Distributions

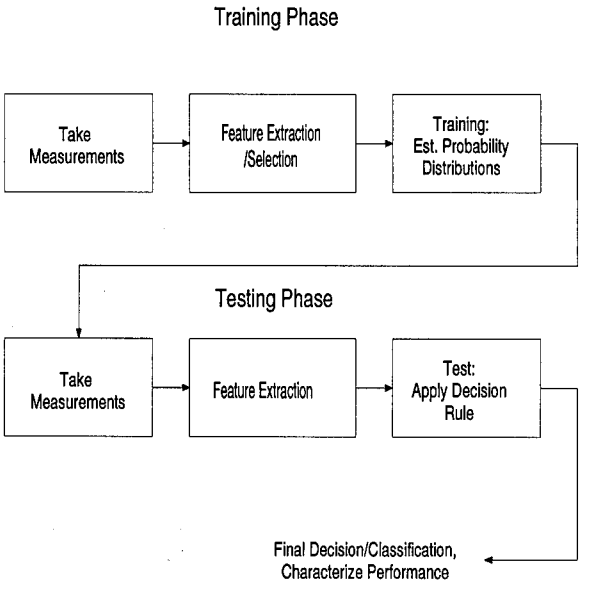

Figure 2.2 Flowchart of Pattern Recognition System

wavelet methods and with Fourier methods for comparison. Clearly, good features lead to good classification, and according to Parsons [11], good features meet the following criteria:

1. Vary widely from class to class.
2. Insensitive to extraneous variables.
3. Stable over long periods of time.
4. Easy to measure.
5. Uncorrelated with other features.

According to Foley [1], if the ratio of training samples per class to feature space dimensionality is less than 3, then a classifier tends to memorize the training data. This indicates, for example, that a feature space dimensionality of ten would need a minimum of 30 training samples to avoid memorizing the training data. Although his work centered on Gaussian data and Gaussian classifiers this rule has become one of the rules of thumb in pattern recognition. We have enough data samples to avoid breaking Foley's rule in this thesis.

Once the features have been chosen, a method of classification is required for the final decision. In statistical pattern recognition, the optimal decision rule is the Bayes decision rule that states the class of the sample in question is the class with the largest *a-priori* probability. As illustrated in figure 2.1, this means the class decision is determined by the higher of the two class conditional distribution curves. In this thesis we use the multilayer perceptron for all classification runs. The multilayer perceptron uses the training data to adjust its weights such that it can approximate a wide range of function classes [2]. Furthermore, it has been shown that the multilayer perceptron approximates the *a-posteriori* class conditional probabilities and hence approximates the Bayes decision optimal decision function [12].

2.4 Summary

In this Chapter we presented a description of the various methods used in the field in the form of a literature review. The following chapter contains the precise mathematical definition

of the methods necessary to implement the wavelet-based feature extraction and classification system.

III. Models

3.1 Introduction

In this Chapter we present the mathematical models used throughout this thesis. An example of every process using a sample from our data set is shown. Included in the introduction is a brief outline of the methods employed in this thesis.

3.1.1 Method For Thesis. The original data consists of signed integer-valued samples of the narrowband signal. We built one classifier to use as a reference using the raw intermediate frequency (IF) pulse data samples.

For all other classifiers we extracted the amplitude and frequency information based on the Double Sideband Suppressed Carrier demodulation algorithm [13]. Three feature extraction and classification systems were considered:

1. Adaptive wavelet representation, classify on weights.
2. Fixed wavelet decomposition, classify on weights.
3. Fixed wavelet decomposition, classify on shifts, dilations and weights.

For each method the cases of amplitude and frequency data were handled separately.

3.1.1.1 Adaptive Wavelet Method – Weights. The wavelet decomposition of a particular signal is used as an initial starting point for the adaptive wavelet representation network–our feature extraction network. Sets of shift and dilation parameters are calculated for each class of data. These sets are combined by union to form the master set of shift and dilation pairs. This master set is used in conjunction with the AWR network to obtain the weight parameters. These parameters are the features that are fed into the multilayer perceptron.

3.1.1.2 Fixed Wavelet Method – Weights. Given either amplitude or frequency data we chose a sample pulse from each class in the training data set. The N wavelets

corresponding to the largest amplitude detail coefficients of the wavelet decomposition of the sample pulses are saved. We then union the saved sets of N wavelets, forming our final set of wavelets for feature extraction. Taking the wavelet decomposition on each pulse in both training and test data sets, we keep only those weights which correspond to the wavelets in our feature extraction set. The weights are the features for the neural network classifier.

3.1.1.3 Fixed Method – Weights, Dilations, Shifts. Given either amplitude or frequency data, we take the wavelet decomposition of each pulse individually and save the triples $(weight, shift, dilation)$ associated with the N largest magnitude detail coefficients. The weights, shifts, and dilations are the features for the neural network classifier.

3.2 Amplitude and Phase Extraction

Figure 3.1 is an example of an IF narrowband signal. We are interested in the amplitude and phase of this signal for use in our classification system. We loosely follow the Double Sideband – Suppressed Carrier demodulation outline given by Stremmler [13]. Consider the representation of a signal

$$s(t) = a(t)\sin(\omega_0 t + \phi(t)),\ t \in \mathbf{R}, \tag{3.1}$$

where ω_0 is the known IF frequency and $a(t)$ and $\phi(t)$ are assumed to be slowly varying amplitude and phase functions, respectively.

If we operate on $s(t)$ with the operators $\mathbf{S}$ and $\mathbf{C}$ defined as multiplication by $\sin(\omega_0 t)$ and $\cos(\omega_0 t)$ respectively and use the trigonometric formulas for $\sin(A+B)$ and $\cos(A+B)$, then we arrive at the following equations:

$$\mathbf{S}s(t) := \frac{a(t)}{2}\left[\cos(\phi(t)) - \cos(2\omega_0 t + \phi(t)\right] \tag{3.2}$$

$$\mathbf{C}s(t) := \frac{a(t)}{2}\left[\sin(\phi(t)) + \sin(2\omega_0 t + \phi(t)\right]. \tag{3.3}$$

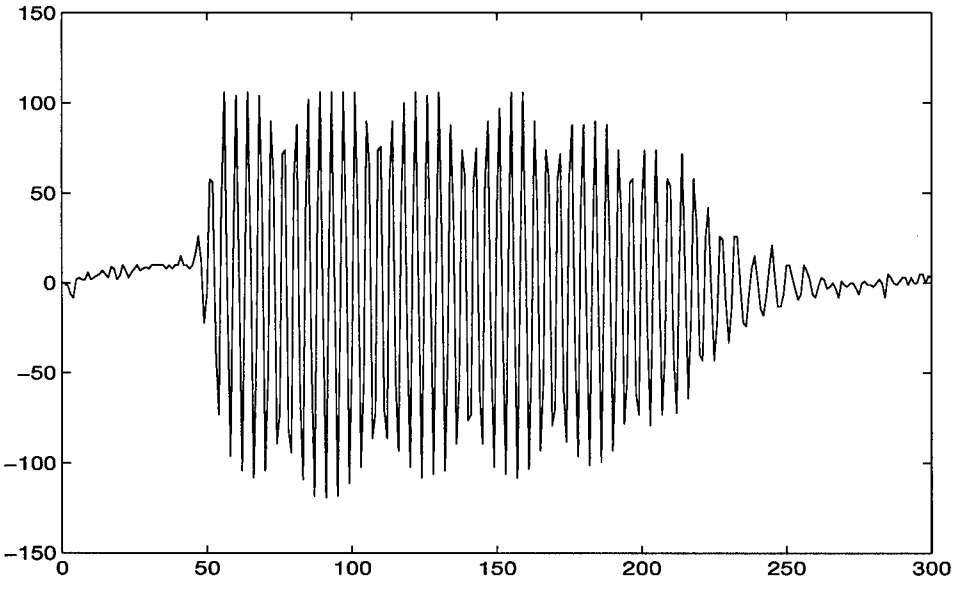

Figure 3.1 A Sample Pulse – Original IF Signal versus Time

Next define the low-pass filter operator $\mathbf{L}$ as multiplication by the characteristic function $\chi(z)$ where $z \in [-\omega_0/2, \omega_0/2]$ and operate on $\mathbf{S}s(t)$ and $\mathbf{C}s(t)$. The result is given in the equations below:

$$x(t) = \mathbf{L}(\mathbf{S}s)(t) = \frac{a(t)}{2}\cos(\phi(t)) \tag{3.4}$$

$$y(t) = \mathbf{L}(\mathbf{C}s)(t) = \frac{a(t)}{2}\sin(\phi(t)). \tag{3.5}$$

From x and y, we obtain

$$a(t) = 2\sqrt{x^2(t) + y^2(t)}, \tag{3.6}$$

and

$$\phi(t) = \arctan\left(\frac{y(t)}{x(t)}\right), \; x(t) \neq 0. \tag{3.7}$$

We then used the following relation to calculate the frequency:

$$\nu(t) = \frac{d\phi(t)}{dt} \tag{3.8}$$

Figures 3.2, 3.3, and 3.4 are calculated from the original signal shown in figure 3.1 and depict the amplitude, phase, and frequency plots respectively.

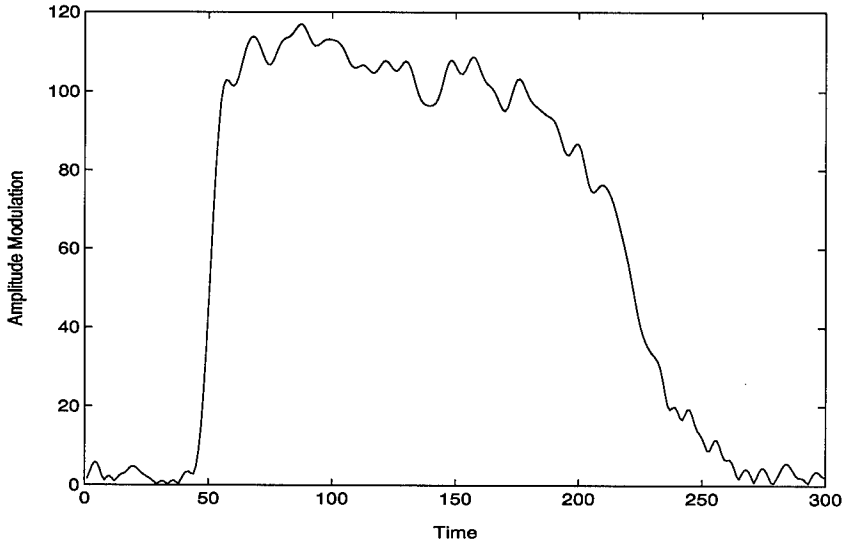

Figure 3.2 A Sample Pulse – Amplitude Modulation versus Time

3.3 Multiresolution Decomposition

In this thesis we implement the multiresolution wavelet decomposition as a quadrature mirror filter (QMF) with downsampling. For a detailed tutorial on wavelet analysis and multiresolution algorithms developed by Mallat [14], consult Smiley [15] or Anderson [16].

3.3.1 Discrete Wavelet Decomposition Using The Daubechies 20-Tap Filter Wavelet.

Since we decided to implement the dyadic wavelet decomposition using the Daubechies 20-tap filter wavelet as a quadrature mirror filter, we high-pass filter the signal at a given resolution, down-sample by a factor of two, and save the resulting detail coefficients. Each

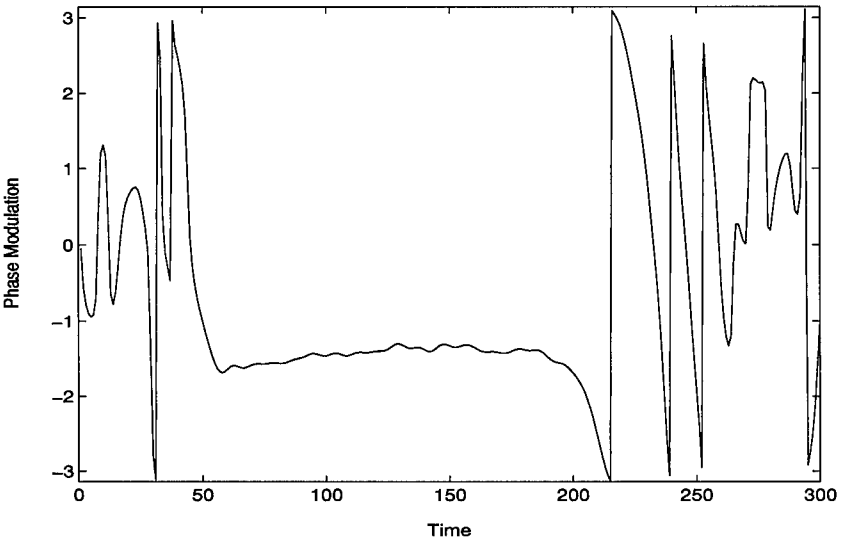

Figure 3.3 A Sample Pulse – Phase Modulation versus Time

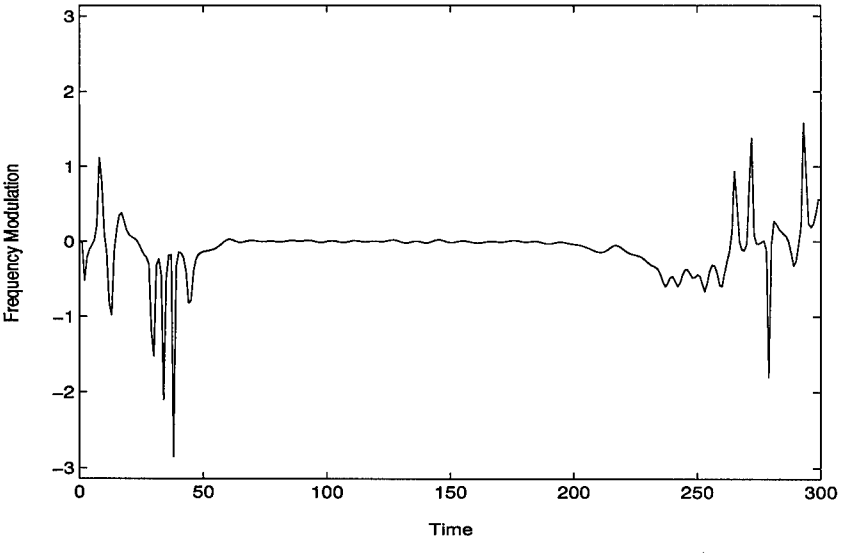

Figure 3.4 A Sample Pulse – Frequency Modulation versus Time

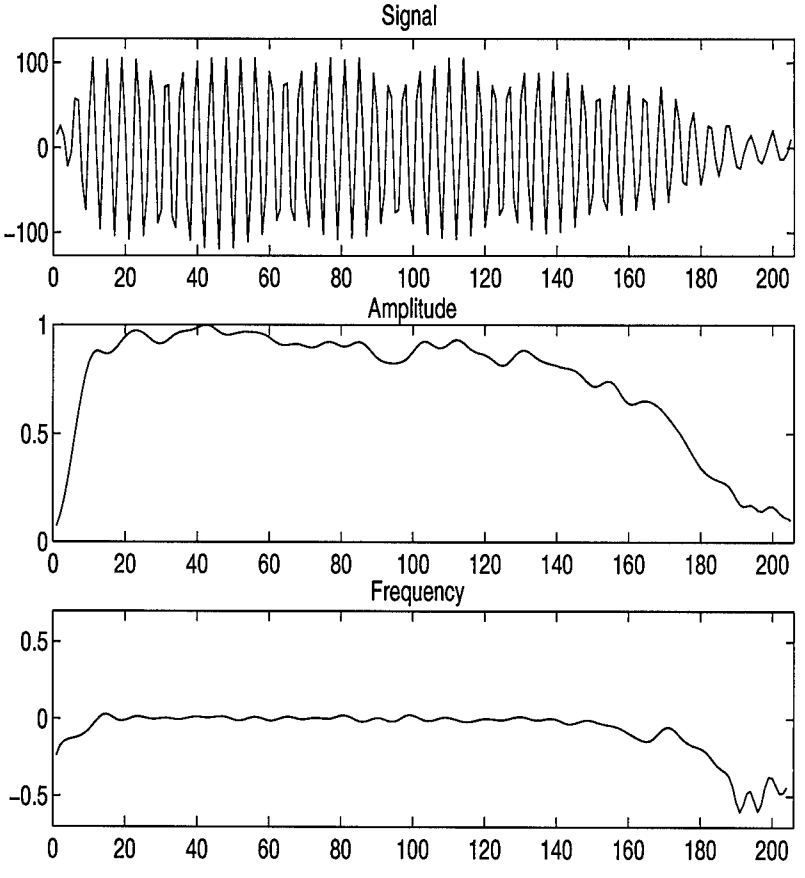

Figure 3.5 Extracted Signal, Amplitude Modulation, and Frequency Modulation versus Time

detail coefficient represents the correlation of the signal with a particular shift of the wavelet at this resolution. Next low-pass filter the signal and again down-sample by a factor of two. The resulting coefficients represent the original signal at a coarser resolution level.

The detail coefficients were sorted in descending order of their magnitude. We then selected a fixed set of wavelets corresponding to the detail coefficients at the top of the list. The list of wavelets is be used by the adaptive wavelet representation network.

First, a multiresolution analysis (MRA) is defined. An MRA is a set of embedded subspaces $V_m \subset \mathbf{L}^2(\mathbf{R})$ such that

$$\cdots \subset V_1 \subset V_0 \subset V_{-1} \subset \cdots. \tag{3.9}$$

These spaces are known as approximation spaces. They satisfy the conditions

$$\bigcap_{m\in\mathbf{Z}} V_m = \{0\} \quad \text{and} \quad \overline{\bigcup_{m\in\mathbf{Z}} V_m} = \mathbf{L}^2(\mathbf{R}). \tag{3.10}$$

Then, with the dilation factor 2,

$$f \in V_m \Longleftrightarrow f(2\cdot) \in V_{m-1}. \tag{3.11}$$

Finally, assume there exists a scaling function $\phi \in V_0$ such that the integer translations of ϕ are orthogonal, and such that $\{\phi_{m,n}\}$ forms a basis for V_m. That is,

$$V_m = \overline{\text{span}\{\phi_{mn}\}_{n\in\mathbf{Z}}}, \tag{3.12}$$

where

$$\phi_{mn}(x) = 2^{-m/2}\phi(2^{-m}x - n). \tag{3.13}$$

Given the above definition, define the detail space W_m as the orthogonal complement of V_m in V_{m-1}. Then

$$W_m \perp V_m, \tag{3.14}$$

$$W_m \subset V_{m-1}, \tag{3.15}$$

and

$$V_m \oplus W_m = V_{m-1}. \tag{3.16}$$

The wavelets are an orthonormal basis for the detail spaces:

$$W_m = \overline{\text{span}\{\psi_{mn}\}_{n \in \mathbf{Z}}}, \tag{3.17}$$

where

$$\psi_{mn}(x) = 2^{-m/2}\psi(2^{-m}x - n). \tag{3.18}$$

The constant $2^{-m/2}$ in equations 3.13 and 3.18 normalizes the energy of the corresponding scaling function or wavelet.

Assume we have the two discrete filters, G and H, which correspond to the scaling function ϕ and the wavelet ψ. Furthermore, assume the two discrete filters G and H satisfy the following relation:

$$g(n) = (-1)^{1-n}h(1-n), \tag{3.19}$$

where g and h are the impulse responses of G and H respectively. Then by definition G is the mirror filter of H. According to Mallat [14], we can calculate the detail coefficients at the current approximation level $m = 1$ by

$$d_{m,k} = \sum_n g(n-2k)c_{m,n}. \tag{3.20}$$

The approximation coefficients for the next level are determined by

$$c_{m+1,k} = \sum_n h(n-2k)c_{m,n}. \tag{3.21}$$

Figure 3.6 depicts the decomposition algorithm with a flowchart diagram. Table 3.1 lists a subset of the detail coefficients sorted by magnitude, along with the corresponding shift and dilation parameters, for the multiresolution wavelet decomposition of the amplitude envelope of the sample signal. Finally, figure 3.7 shows the data structure employed in the fast decomposition algorithm.

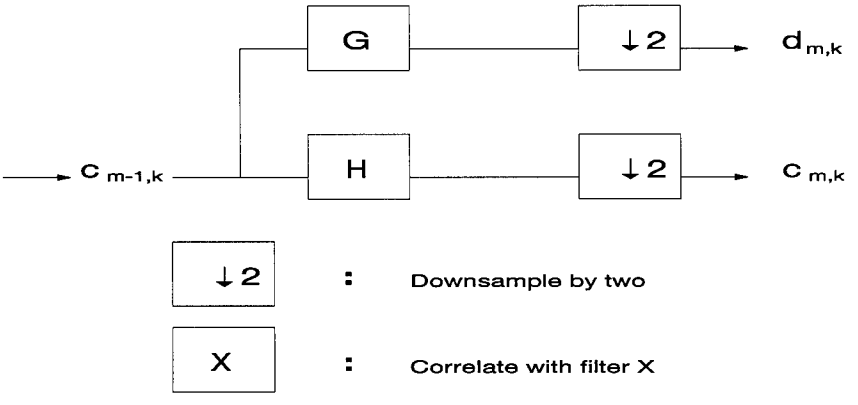

Figure 3.6 Diagram of Filtering Algorithm Representing One Level of a Wavelet Decomposition.

3.3.2 Daubechies 20-Tap Wavelet Properties. Shown in figure 3.8 are the impulse responses of the two filters described in the previous section for the Daubechies 20-tap wavelet. Table 3.2 lists the filter coefficients.

3.4 Wavelet Neural Networks

In this thesis we used Szu's Adaptive Wavelet Representation (AWR) network to give a representative set of wavelets for feature extraction purposes. Figure 3.9 depicts the AWR network.

Table 3.1 Largest 25 Detail Coefficients in Magnitude and Their Corresponding Shift and Dilation Parameters from the Wavelet Decomposition of the Amplitude Envelope of the Sample Pulse

	Weight	*Dilation*	*Shift*
1	-5.305	256	0
2	-1.427	128	0
3	-1.376	64	0
4	1.125	64	64
5	-1.071	64	128
6	-0.7982	128	128
7	-0.6201	32	0
8	0.4249	32	32
9	-0.4222	32	224
10	-0.3694	16	0
11	0.3572	16	16
12	-0.3183	32	64
13	-0.2196	16	32
14	0.1909	8	8
15	-0.1803	8	0
16	0.1418	16	112
17	-0.1414	8	16
18	0.1301	32	128
19	0.1282	32	192
20	0.1209	16	224
21	-0.1147	16	192
22	0.1008	16	176
23	-0.09548	16	160
24	-0.08711	8	192
25	0.08438	16	48

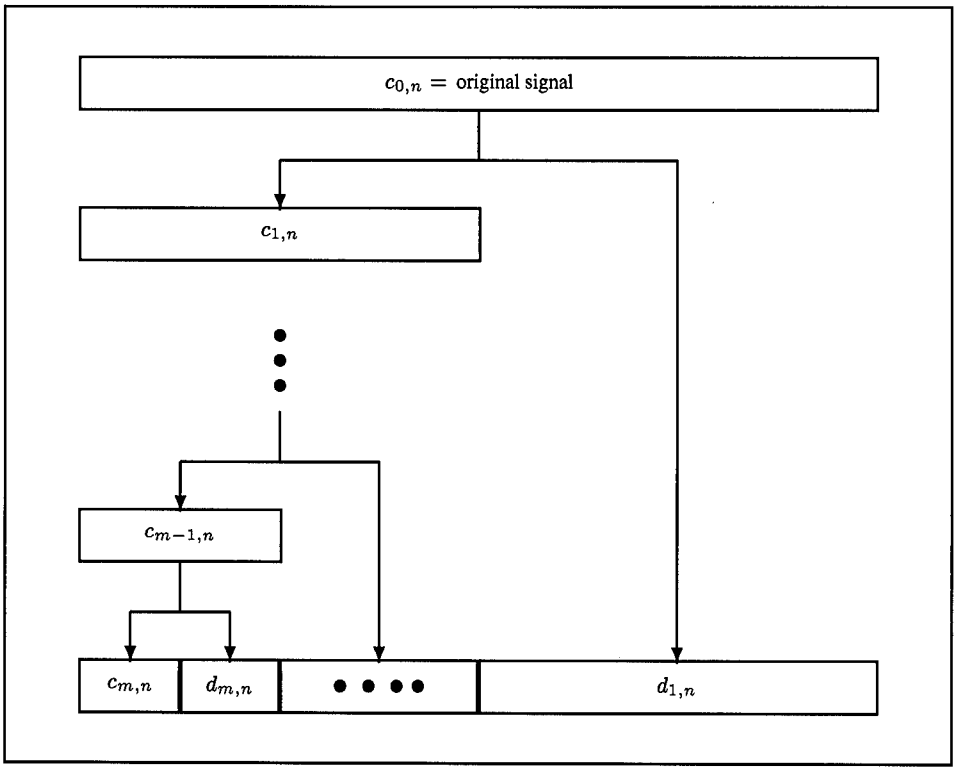

Figure 3.7 Data Structure for Fast Wavelet Decomposition Algorithm

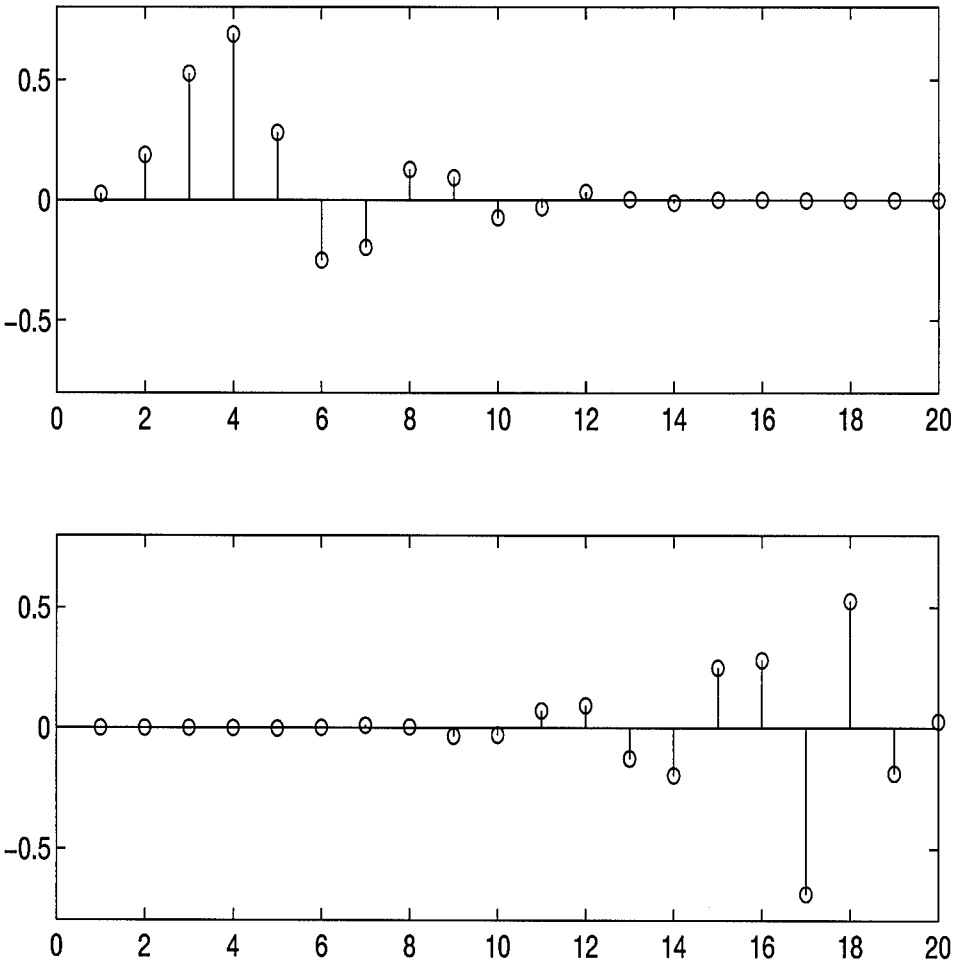

Figure 3.8 Scaling Filter H (top) and Wavelet Filter G (bottom) for the Daubechies 20-Tap Filter Wavelet

Table 3.2 Filter Coefficients for the Daubechies 20-Tap Wavelet

Scaling Filter H	*Wavelet Filter G*
2.6670058e-02	1.3264203e-05
1.8817680e-01	9.3588670e-05
5.2720119e-01	1.1646686e-04
6.8845904e-01	-6.8585670e-04
2.8117234e-01	-1.9924053e-03
-2.4984642e-01	1.3953517e-03
-1.9594627e-01	1.0733175e-02
1.2736934e-01	3.6065536e-03
9.3057365e-02	-3.3212674e-02
-7.1394147e-02	-2.9457537e-02
-2.9457537e-02	7.1394147e-02
3.3212674e-02	9.3057365e-02
3.6065536e-03	-1.2736934e-01
-1.0733175e-02	-1.9594627e-01
1.3953517e-03	2.4984642e-01
1.9924053e-03	2.8117234e-01
-6.8585670e-04	-6.8845904e-01
-1.1646686e-04	5.2720119e-01
9.3588670e-05	-1.8817680e-01
-1.3264203e-05	2.6670058e-02

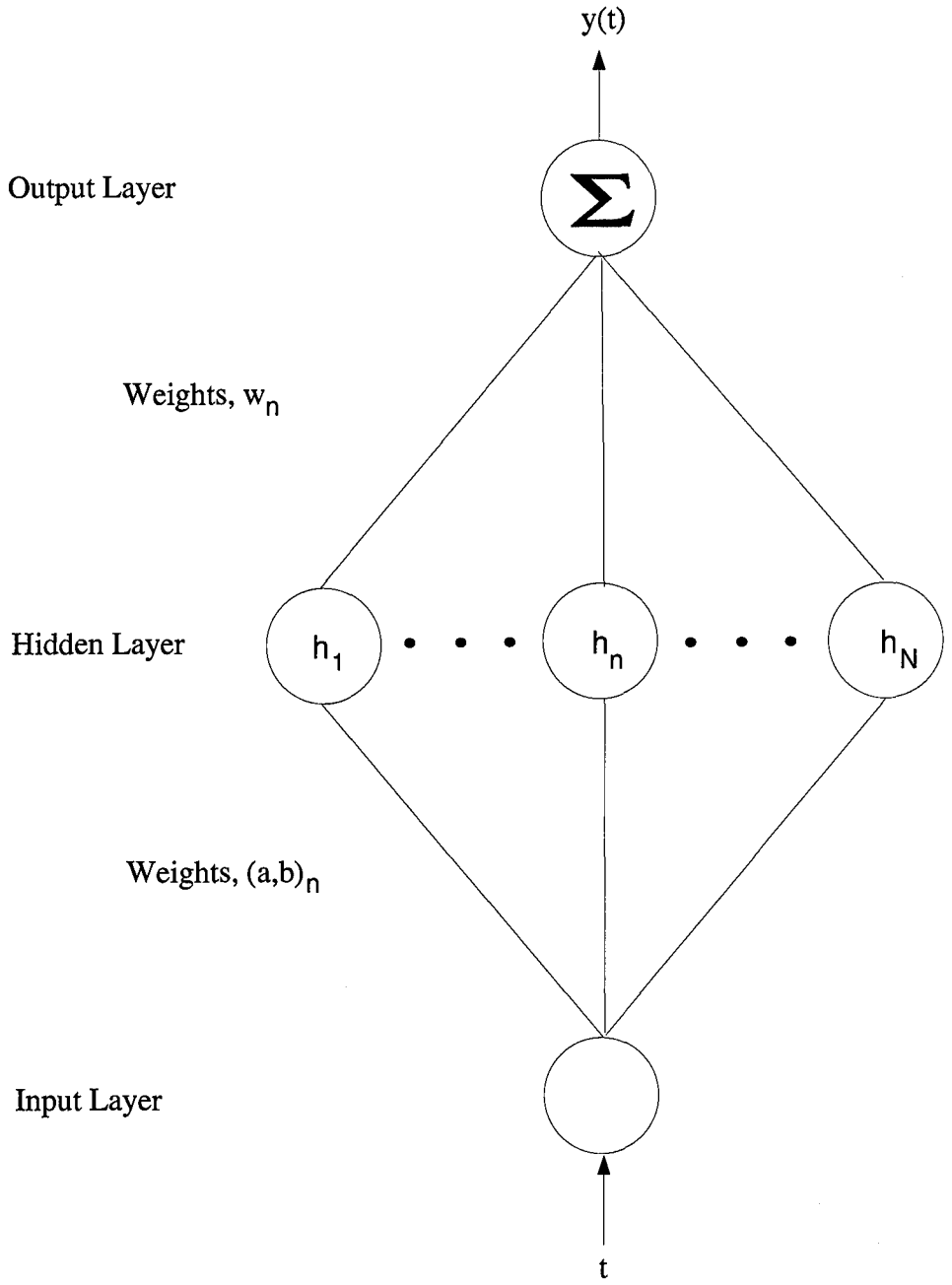

Figure 3.9 Adaptive Wavelet Representation Network Diagram

3.4.1 Szu, et al – AWR. Given an list of candidate wavelets obtained from the discrete wavelet decomposition, we are interested in optimizing the function:

$$y(t) = \sum_{n=1}^{N} w_n h(\frac{t-b_n}{a_n});\ t = 1, \ldots, T;\ w_n, b_n \in \mathbf{R};\ a_n \in \mathbf{R} - \{0\} \tag{3.22}$$

where $h \in \mathbf{L}^2(\mathbf{R})$ is a wavelet and $\{(t, s(t))\}_{t=1}^{T}$ is the training data set. The free parameters to be determined in equation 3.22 are $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{w}$, where:

$$\mathbf{a} := \left(a_1,\ a_2,\ \ldots,\ a_n,\ \ldots,\ a_N \right)^T,$$
$$\mathbf{b} := \left(b_1,\ b_2,\ \ldots,\ b_n,\ \ldots,\ b_N \right)^T,$$
$$\mathbf{w} := \left(w_1,\ w_2,\ \ldots,\ w_n,\ \ldots,\ w_N \right)^T.$$

Furthermore, define

$$\mathbf{y} := \left(y(1),\ y(2),\ \ldots,\ y(t),\ \ldots,\ y(T) \right)^T,$$
$$\mathbf{s} := \left(s(1),\ s(2),\ \ldots,\ s(t),\ \ldots,\ s(T) \right)^T,$$

and

$$\mathbf{h}(t) := \left(h(\tfrac{t-a_1}{b_2}),\ h(\tfrac{t-a_2}{b_2}),\ \ldots,\ h(\tfrac{t-a_n}{b_n}),\ \ldots,\ h(\tfrac{t-a_N}{b_N}) \right)^T.$$

We want to minimize the functional

$$E = 1/2 \sum_{t=1}^{T} (s(t) - y(t))^2. \tag{3.23}$$

We choose to minimize E using the gradient descent minimization algorithm for the variables $\mathbf{a}$ and $\mathbf{b}$. Therefore we must find the partial derivatives of equation 3.23 with respect to $\mathbf{a}$ and $\mathbf{b}$. We see that for $n = 1, \ldots, N$

$$\frac{\partial E}{\partial a_n} = \sum_{t=1}^{T} (y(t) - s(t)) w_n h'(\frac{t-b_n}{a_n}) \left(\frac{t-b_n}{a_n^2} \right), \tag{3.24}$$

where the prime indicates the derivative of the function h, and

$$\frac{\partial E}{\partial b_n} = \sum_{t=1}^{T}(y(t) - s(t))w_n h'(\frac{t-b_n}{a_n})\left(\frac{1}{a_n^2}\right). \tag{3.25}$$

The resulting update for $n = 1, \ldots, N$ is as follows:

$$a_n^{\text{new}} = a_n^{\text{old}} - \eta\frac{\partial E}{\partial a_n}, \tag{3.26}$$

$$b_n^{\text{new}} = b_n^{\text{old}} - \eta\frac{\partial E}{\partial b_n}, \tag{3.27}$$

where $\eta \in \mathbf{R}$ is the stepsize parameter of the gradient descent update.

Since the error functional E is quadratic in terms of the weights $\mathbf{w}$ we can solve for the optimal $\mathbf{w}$ analytically. Consider

$$y(t) = \mathbf{h}^T(t)\mathbf{w}, \ \forall\, t = 1, 2, \ldots, T \tag{3.28}$$

and define

$$\mathbf{H} = \left[\ \mathbf{h}(1)\ \middle|\ \mathbf{h}(2)\ \middle|\ \cdots\ \middle|\ \mathbf{h}(t)\ \middle|\ \cdots\ \middle|\ \mathbf{h}(T)\ \right]. \tag{3.29}$$

We want to minimize

$$\|\mathbf{s} - \mathbf{H}^T\mathbf{w}\|^2, \tag{3.30}$$

where $\|\cdot\|$ is the Euclidean norm. The general solution to this optimization problem is given by

$$\mathbf{H}\mathbf{H}^T\mathbf{w} = \mathbf{H}\mathbf{s}. \tag{3.31}$$

If $\mathbf{HH}^T$ is invertible, then we can solve for $\mathbf{w}$ by multiplying both sides of equation 3.30 by $(\mathbf{HH}^T)^{-1}$ resulting in the following expression for $\mathbf{w}$:

$$\mathbf{w} = (\mathbf{HH}^T)^{-1}\mathbf{Hs}. \tag{3.32}$$

If $\mathbf{HH}^T$ is not invertible because the matrix has less than full rank, then we have many solutions to equation 3.30. In this case we choose the $\mathbf{w}$ with the minimum Euclidean norm.

We return to the sample pulse. Using the wavelets that correspond to the 20 largest magnitude detail coefficients from the discrete wavelet decomposition in Section 3.3.1 as initial starting points for the shift, dilation and weight parameters $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{w}$, the adaptive wavelet representation of the sample amplitude envelope are computed. Figure 3.10 shows the original amplitude signal and the resulting approximation using 20 adaptive wavelets. Table 3.3 lists the final shift and dilation parameters after 21 training epochs of the AWR network on the amplitude sample pulse, where an epoch is defined as one pass through the training data.

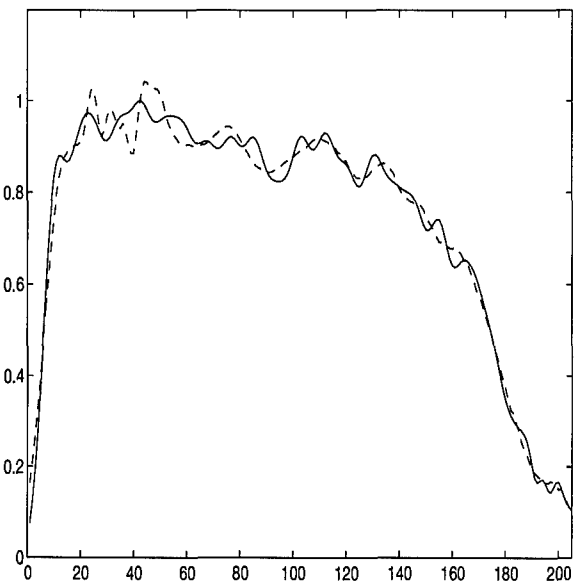

Figure 3.10 Adaptive Wavelet Representation (dashed) and Amplitude Envelope of Sample (solid) Pulse Using 20 Adaptive Wavelets

Table 3.3 Shift and Dilation Parameters from the Adaptive Wavelet Representation Network for a Sample Pulse

Dilation	*Shift*
270.813	26.7246
127.395	2.11566
63.5368	-0.337342
63.5751	67.3347
65.6245	126.267
136.675	118.726
30.4872	0.0877888
31.7195	32.2841
32.0804	236.239
15.8336	0.983875
13.2896	15.1990
30.7247	64.0980
15.6767	33.2808
8.66702	10.1273
10.4481	3.69206
15.7139	112.395
7.18928	16.1474
30.5117	126.889
32.3301	187.669
16.0802	224.238

3.5 *Multilayer Perceptrons*

The Multilayer Perceptron Network performs classifications by partitioning the feature space into regions of interest, grouping patterns from the same class via linear decision functions, $d(\mathbf{X}) \in \mathbf{R}$ where

$$d(\mathbf{X}) = W_1X_1 + W_2X_2 + \ldots + W_{N-1}X_{N-1} + W_N,\ \mathbf{X} \in \mathbf{R}^{N-1}, \mathbf{W} \in \mathbf{R}^N. \quad (3.33)$$

In a multidimensional feature space, $d(\mathbf{X})$ can be positioned such that any pattern vector, $\mathbf{X}$, belonging to one class yields a positive quantity when the features are substituted into $d(\mathbf{X})$ while any pattern belonging to another class yields a negative quantity.

The characteristics of the linear decision functions can be modeled by nodes (figure 3.11) with sigmoidal activation functions

$$y = f(\mathbf{X}) = \left[\frac{1}{1 + e^{-\sum_{n=1}^{N-1} X_n W_n + W_N}}\right], \quad (3.34)$$

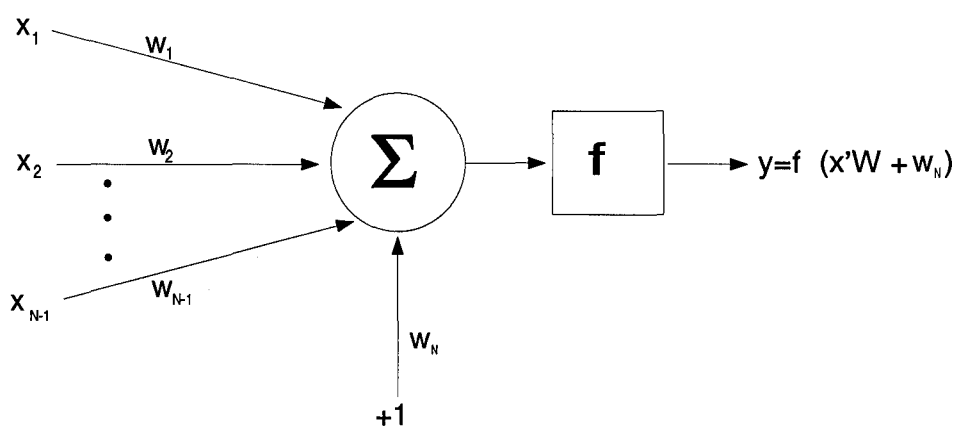

Figure 3.11 One Node with Sigmoidal Activation

The resulting network structure can be seen in Figure 3.12. Each input is weighted and then the weighted inputs are summed at the nodes in the hidden layer and the bias term

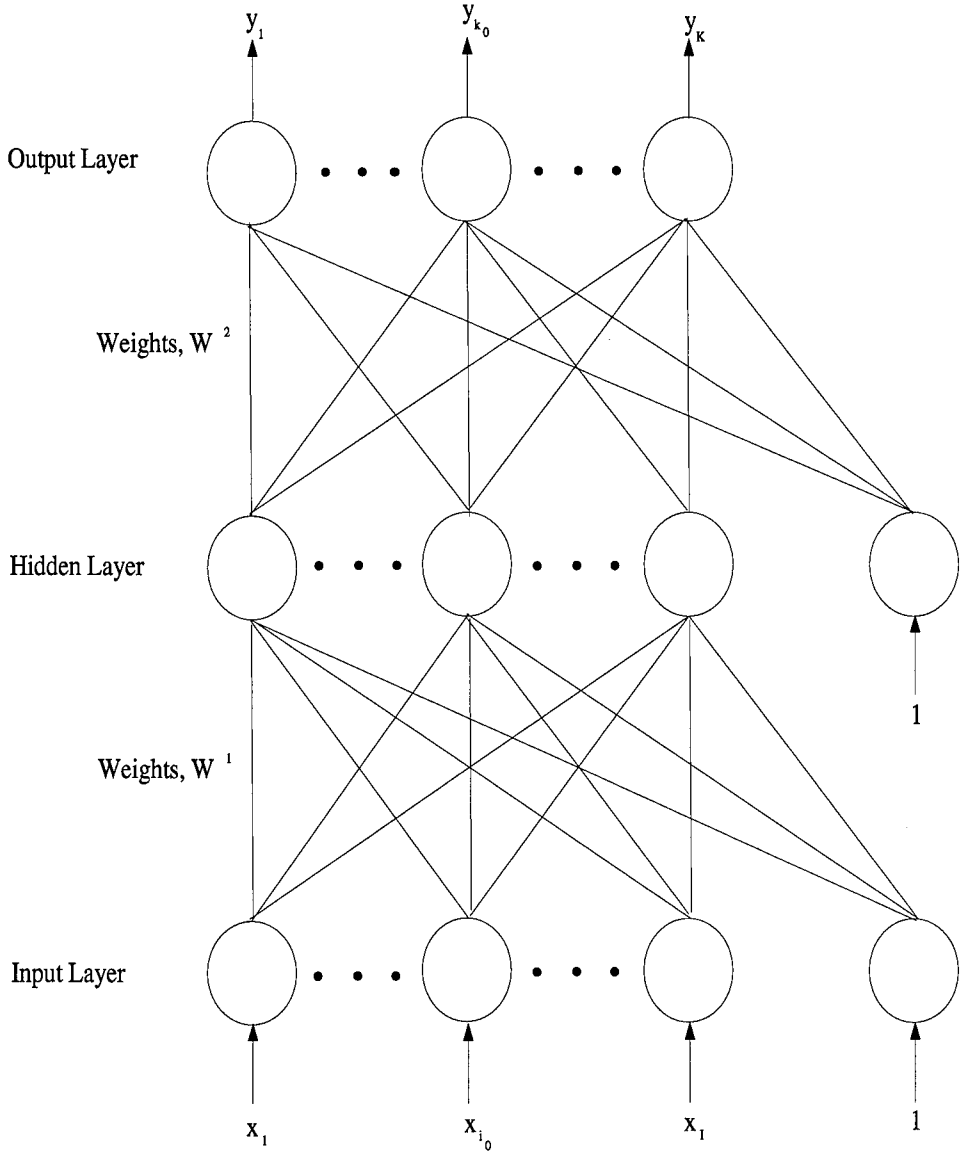

Figure 3.12 Multilayer Perceptron

$X_{I+1} = 1$ is added. This bias term is added because without it the decision functions all must pass through the origin. Figure 3.13 depicts an example two-class problem which could not be solved with a multilayer perceptron without a bias term.

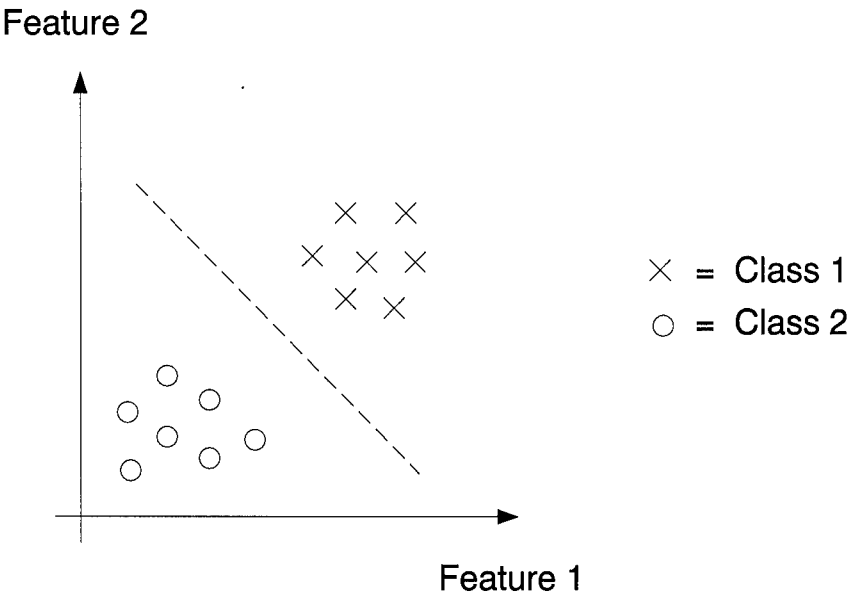

Figure 3.13 Two-Class Example which Demonstrates the Need for a Bias Term in a Multilayer Perceptron

For training we want to minimize the error functional

$$E = \frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2, \tag{3.35}$$

where d_k is the desired output and y_k is the actual output. The weights must be updated using gradient descent minimization algorithm. The generalized learning law is shown below:

$$W^+ = W^- - \eta\frac{\partial E}{\partial W}, \tag{3.36}$$

where, W^+ is the updated weight, W^- is the old weight, and $\eta \in \mathbf{R}^+$ is a constant. The learning law is derived below after the output is defined for the sigmoid activation function.

$$y_{k_0} = f(\tilde{x}) = \frac{1}{1+e^{-\tilde{x}}}, \tag{3.37}$$

where

$$\tilde{x} = \sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2 \tag{3.38}$$

3.5.1 Derivation. For the weights between the output layer and the hidden layer we have

$$W_{j_0k_0}^{2+} = W_{j_0k_0}^{2-} - \eta \frac{\partial E}{\partial W}. \tag{3.39}$$

Then, analyzing the partial derivative term in equation 3.39 yields the following:

$$\begin{aligned}
\frac{\partial E}{\partial W_{j_0k_0}^2} &= \frac{\partial}{\partial W_{j_0k_0}^2}\{\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\} \\
&= \frac{\partial}{\partial W_{j_0k_0}^2}\frac{1}{2}\{(d_1 - y_1)^2 + \cdots + (d_{k_0}^2 - y_{k_0})^2 + \cdots + (d_k - y_k)^2\} \\
&= (d_{k_0} - y_{k_0})(-1)\frac{\partial y_{k_0}}{\partial W_{j_0k_0}^2} \\
&= -(d_{k_0} - y_{k_0})\frac{\partial}{\partial W_{j_0k_0}^2}(1 + e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})^{-1} \\
&= -(d_{k_0} - y_{k_0})(-1)(1 + e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})^{-2}(e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})\frac{\partial}{\partial W_{j_0k_0}^2}(-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2) \\
&= -(d_{k_0} - y_{k_0})\frac{e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2}}{(1 + e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})^2}(x_{j_0}^2) \\
&= -(d_{k_0} - y_{k_0})(y_{k_0})(1 - y_{k_0})(x_{j_0}^2),
\end{aligned}$$

and therefore the update rule is given by:

$$W_{j_0k_0}^{2+} = W_{j_0k_0}^{2-} + \eta(d_{k_0} - y_{k_0})(y_{k_0})(1 - y_{k_0})(x_{j_0}^2). \tag{3.40}$$

Consider the weights between the input layer and the hidden layer:

$$W_{i_0j_0}^{1+} = W_{i_0j_0}^{1-} - \eta \frac{\partial E}{\partial W_{i_0j_0}^1}. \tag{3.41}$$

Again, evaluate the partial derivative term of the above equation 3.41 :

$$\begin{aligned}
\frac{\partial E}{\partial W_{i_0j_0}^1} &= \frac{\partial}{\partial W_{i_0j_0}^1}\{\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\} \\
&= -\sum_{k=1}^{K}(d_k - y_k)\frac{\partial y_k}{\partial W_{i_0j_0}^1} \\
&= -\sum_{k=1}^{K}(d_k - y_k)\frac{\partial}{\partial W_{i_0j_0}^1}(1 + e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})^{-1} \\
&= -\sum_{k=1}^{K}(d_k - y_k)(y_k)(1 - y_k)\frac{\partial}{\partial W_{i_0j_0}^1}(-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2) \\
&= -\sum_{k=1}^{K}(d_k - y_k)(y_k)(1 - y_k)(-W_{j_0k}^2)\frac{\partial}{\partial W_{i_0j_0}^1}(x_{j_0}^2) \\
&= -\sum_{k=1}^{K}(d_k - y_k)(y_k)(1 - y_k)(-W_{j_0k}^2)(x_{j_0}^2)(1 - x_{j_0}^2)(-x_{i_0}^1),
\end{aligned}$$

and therefore:

$$W_{i_0j_0}^{1+} = W_{i_0j_0}^{1-} + \eta\sum_{k=1}^{K}(d_k - y_k)(y_k)(1 - y_k)(W_{j_0k}^2)(x_{j_0}^2)(1 - x_{j_0}^2)(x_{i_0}^1). \tag{3.42}$$

3.6 Summary

In this chapter we have presented the mathematical methods necessary to implement our wavelet based feature extraction and classification system. We have shown how to demodulate a narrowband signal, wavelet decompose and optimize an adaptive wavelet representation of

the signal, and update the weights of a multilayer perceptron. In Chapter IV we present our results using these methods.

IV. Implementations and Results

4.1 Introduction

All results were obtained by performing cross-validation testing on the original data files. Three data files for each of four classes of signals were given. For each class the files were supposedly obtained from the same source. The data was therefore split into training and testing sets by assigning two of three data files to the training set and the remaining file to the testing set. All three permutations make up the complete cross-validation test suite (see Table 4.1).

Table 4.1 Data Sets for Cross-Validation Testing per Class ($i = 1, 2, 3, 4$)

Permutation	*Training Data*	*Testing Data*
1	File i_1 File i_2	File i_3
2	File i_1 File i_3	File i_2
3	File i_2 File i_3	File i_1

Figures 4.1, 4.2, 4.3, and 4.4 show a sample for each class from our data set. Each figure displays the original signal along with its demodulated amplitude and frequency. To register the data we normalize the amplitude envelope to a unit maximum, determine the half amplitude point of the amplitude graph of the pulse, backtrack five samples, and then extract enough samples so as to have a vector which encompasses the signal with a few samples of noise at either end. The total number of samples extracted was 205. Figures 4.5 and 4.6 show the the demodulated amplitude and frequency signals overlayed for all four classes.

4.2 Reference Experiments

Results are presented for three reference experiments in this section. The classification was performed on the original narrowband IF signal, its amplitude modulation, and on its frequency modulation.

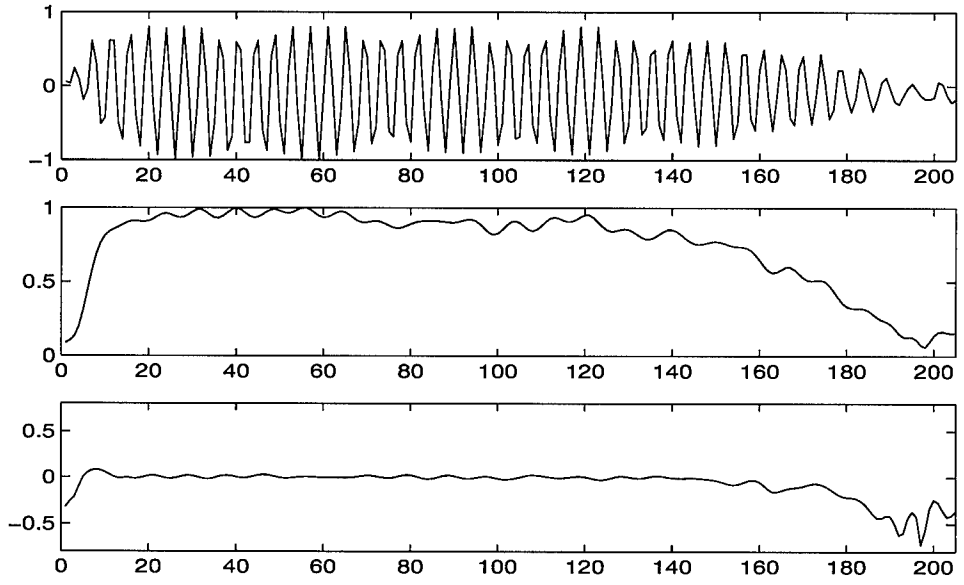

Figure 4.1 Class 1 Sample: Signal, Amplitude Modulation, Frequency Modulation versus Time

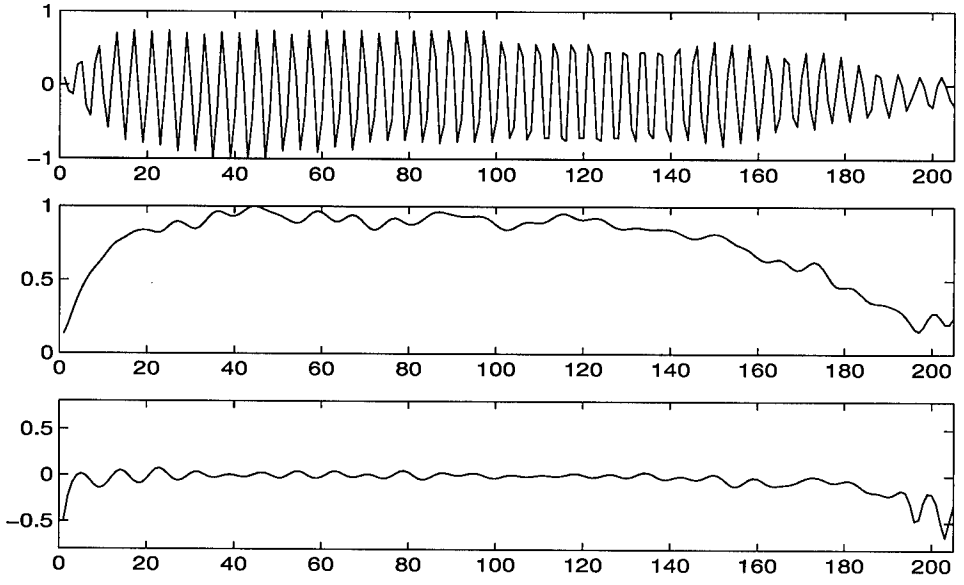

Figure 4.2 Class 2 Sample: Signal, Amplitude Modulation, Frequency Modulation versus Time

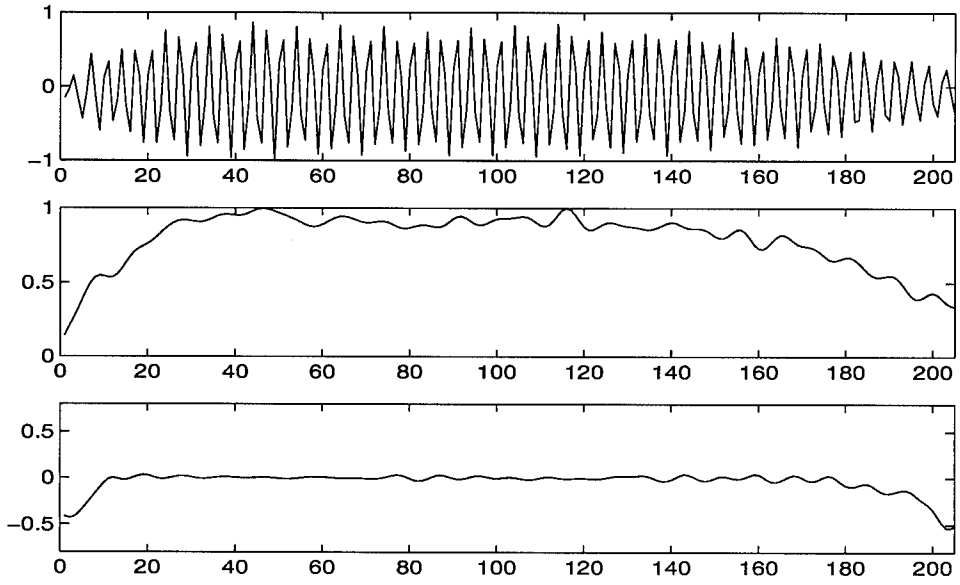

Figure 4.3 Class 3 Sample: Signal, Amplitude Modulation, Frequency Modulation versus Time

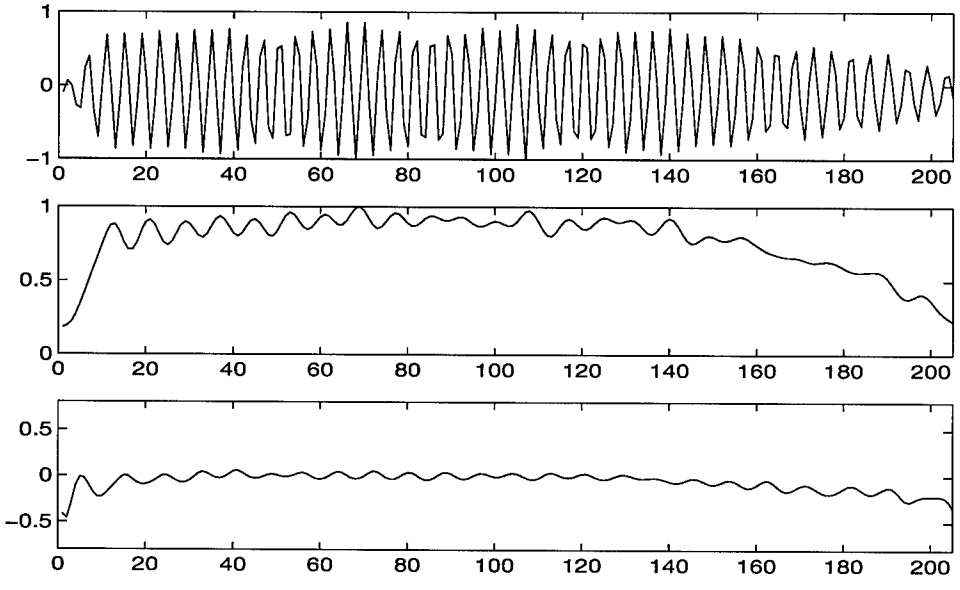

Figure 4.4 Class 4 Sample: Signal, Amplitude Modulation, Frequency Modulation versus Time

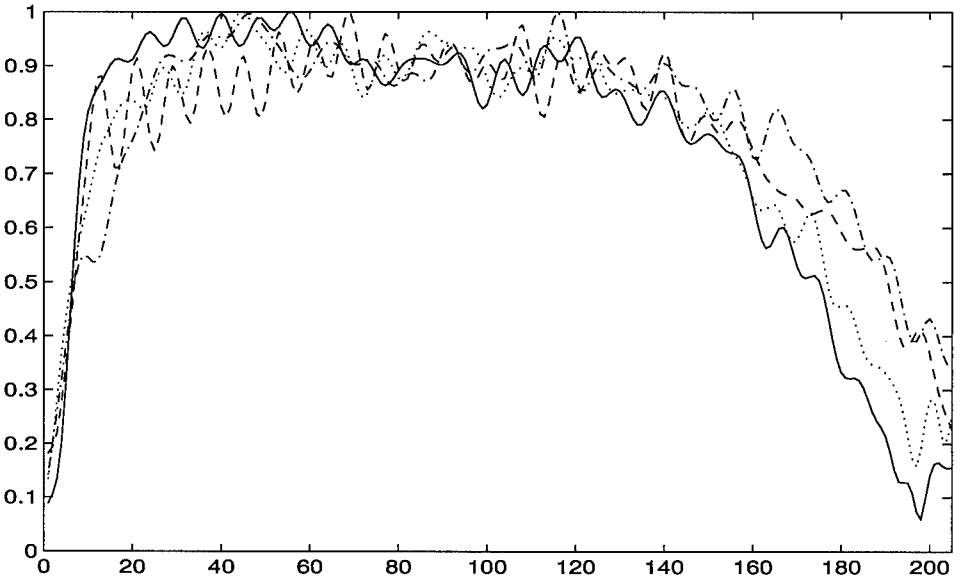

Figure 4.5 One Sample from Each Class: Amplitude Modulation versus Time

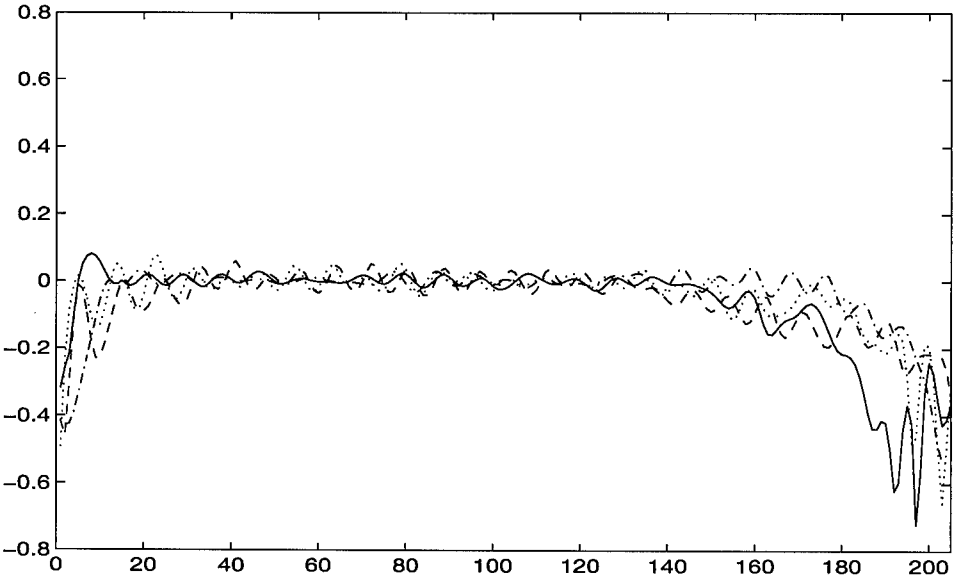

Figure 4.6 One Sample from Each Class: Frequency Modulation versus Time

4.2.1 Original Data. Results for the original raw data are obtained from a network with 205 inputs and 100 hidden nodes. Table 4.2 shows the confusion matrix and classification percentages of the data set under cross-validation testing. The results obtained on the original data are an indication of what is possible for this data set. Note that it is not always feasible to classify based on all of the original data. This test is included here since the sample length of a pulse was relatively short and it was thus possible to train a classifier on the original data.

Table 4.2 Original Data, 100 Hidden Nodes: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	272	28		
2	20	280		1
3	10	12	308	2
4			2	368

Class	*Patterns*	*Errors*	*Percent*
1	300	28	9.3
2	300	21	6.3
3	332	24	7.2
4	370	2	0.5
Testing Error	1302	75	5.8
Training Error	2604	108	4.2

Networks with 205 inputs and 25 hidden nodes were also considered. Table 4.3 shows the confusion matrix and classification percentages of the original data set under cross-validation testing. These tables are included because the input node analysis presented in section 4.9 revealed that 25 hidden nodes resulted in the best classifier in this particular case.

4.2.2 Amplitude Data. Results for the amplitude envelope data are obtained by demodulating the original signal, keeping only amplitude information, and classifying the raw amplitude information with a network of 205 inputs and 24 hidden nodes. Table 4.4 shows the

Table 4.3 Original Data, 25 Hidden Nodes: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	280	19	1	
2	27	273		
3	9	13	308	
4			2	367

Class	*Patterns*	*Errors*	*Percent*
1	300	20	6.7
2	300	27	9.0
3	332	22	6.6
4	370	2	0.5
Testing Error	1302	69	5.3
Training Error	2604	94	3.7

confusion matrix and classification percentages of the data set under cross-validation testing. The results obtained on the amplitude data are an indication of what is possible for this data set. Note that it is not always feasible to classify on the raw amplitude data due to feature vector dimensionality considerations.

4.2.3 Frequency Data. Results for the frequency data are obtained by demodulating the original signal, keeping only frequency information, and classifying on the raw frequency information with a network of 205 inputs and 24 hidden nodes. Table 4.5 shows the confusion matrix and classification percentages of the data set under cross-validation testing. The results obtained on the frequency data are an indication of what is possible for this data set. Note that it is not always feasible to classify on the raw frequency data due to feature vector dimensionality considerations.

Table 4.4 Amplitude Data, 24 Hidden Nodes: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	290	8	1	1
2	24	275	1	1
3	9	12	311	
4			4	366

Class	*Patterns*	*Errors*	*Percent*
1	300	10	3.3
2	300	26	8.7
3	332	21	6.3
4	370	4	1.1
Testing Error	1302	61	4.7
Training Error	2604	92	3.5

Table 4.5 Frequency Data, 24 Hidden Nodes: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	291	6	2	1
2	17	281	2	
3	9	12	311	
4			3	367

Class	*Patterns*	*Errors*	*Percent*
1	300	9	3.0
2	300	19	6.3
3	332	21	6.3
4	370	3	0.8
Testing Error	1302	52	4.0
Training Error	2604	96	3.7

4.3 *Fourier Transform – Weights*

Since the Fourier transform is the *de facto* standard signal processing tool, classification results using the Fourier transform to extract features for classification are presented in this section.

4.3.1 Original. Using the original narrowband IF signal data a classifier was built with coefficients of the Fourier transform as features. There were 27 coefficients of interest centered about ω_0. Thus, 54 input nodes were obtained by treating the real and imaginary parts of the 27 tupels obtained from the complex Fourier coefficients as individual inputs. Table 4.6 shows the results for a network with 30 hidden nodes.

Table 4.6 Fourier Coefficient Features, Original IF Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	266	27	7	
2	11	280	8	1
3		20	310	2
4			4	366

Class	*Patterns*	*Errors*	*Percent*
1	300	34	11.3
2	300	20	6.7
3	332	22	6.6
4	370	4	1.1
Testing Error	1302	80	6.1
Training Error	2604	97	3.7

4.3.2 Amplitude. Using the amplitude envelope information extracted from the original narrowband signal data a classifier was built with low frequency coefficients of the Fourier transform. The zero-frequency (DC) coefficient was discarded and the first 27 positive frequency Fourier coefficients were saved, corresponding roughly to the low-pass filter used

for extracting the amplitude envelope from the narrowband signal. Thus, 54 input nodes are obtained by treating the real and imaginary parts of the 27 tupels obtained from the complex Fourier coefficients as individual inputs. Table 4.7 shows the results for a network with 30 hidden nodes.

Table 4.7 Low Frequency Fourier Coefficient Features, Amplitude Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	292	3	2	1
2	26	279	1	
3	8	13	311	
4			2	368

Class	*Patterns*	*Errors*	*Percent*
1	300	6	2.0
2	300	27	9.0
3	332	21	6.3
4	370	2	0.5
Testing Error	1302	56	4.3
Training Error	2604	96	3.7

4.3.3 Frequency. Results are presented in this section for a classifier whose feature vectors consist of the low frequency Fourier coefficients obtained from the frequency signal extracted from the original pulses. The 54 input nodes were obtained in the same manner as for the amplitude data in section 4.3.2. Table 4.8 shows the results for a network with 18 hidden nodes.

4.4 Adaptive Wavelet Features – Weights

Results are presented in this section for classifying on the weights generated by the adaptive wavelet representation algorithm. The choice of weights is determined by unioning the sets of adaptive wavelets which correspond to each class. These sets were generated by

Table 4.8 Low Frequency Fourier Coefficient Features, Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	292	6	2	
2	19	279	2	
3	8	14	310	
4		1	2	367

Class	*Patterns*	*Errors*	*Percent*
1	300	8	2.7
2	300	21	7.0
3	332	22	6.6
4	370	3	0.8
Testing Error	1302	54	4.2
Training Error	2604	98	3.8

using the 20 wavelets which had the largest weights (in absolute value) from a sample pulse as starting points for the Adaptive Wavelet Representation algorithm. The union of the four sets of 20 wavelets resulted in a set of 80 wavelets which was used to calculate the features, that is, the weights.

Note, it was determined that the simple union is not necessarily optimal. In an experiment the number of wavelets was reduced by 20% by averaging wavelets which had nearly equal shift and dilation parameters; within 1% of each other. The new classifier was able to produce slightly better results on the same data. However, this area was left for future research as the intent in this thesis is to demonstrate the concept of adaptive feature extraction.

A simple method of reducing the number of features is to start with fewer nodes per class in the adaptive wavelet representation network. This was implemented for 15, 10, 5, and 3 nodes per class. Results for 5 and 3 nodes are included below.

As can be seen from the results in this section, there is a clear advantage to using frequency features in the adaptive wavelet case. Future research should be directed towards determining by how much the compression ratio and the performance can be improved simultaneously and what the tradeoffs are at the limits of both performance and compression ratio.

4.4.1 Amplitude Features – 80 Total Nodes. Results for the amplitude data are obtained from a network with 80 input and 20 hidden nodes. Table 4.9 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

Table 4.9 Adaptive Wavelet Features, 80 Features Total, Amplitude Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	248	24	9	19
2	36	245	7	12
3	13	17	302	
4	18	19	3	330

Class	*Patterns*	*Errors*	*Percent*
1	300	52	17.3
2	300	55	18.3
3	332	30	9.0
4	370	40	10.8
Testing Error	1302	177	13.6
Training Error	2604	113	4.3

4.4.2 Frequency Features – 80 Total Nodes. Results for the amplitude data are obtained from a network with 80 input and 20 hidden nodes. Table 4.10 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

Table 4.10 Adaptive Wavelet Features, 80 Features Total, Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	275	15	6	4
2	25	262	1	12
3	10	14	306	2
4	3	10	5	352

Class	*Patterns*	*Errors*	*Percent*
1	300	25	8.3
2	300	38	12.7
3	332	26	7.8
4	370	18	4.9
Testing Error	1302	107	8.2
Training Error	2604	95	3.7

4.4.3 Amplitude Features – 20 Total Nodes. Results for the amplitude data are obtained from a network with 20 input and 15 hidden nodes. Table 4.11 shows the confusion matrix and classification percentages of the data set under cross-validation testing. By selecting fewer nodes per class, there was greater relative sum-squared error in the adaptive representation networks. However, the goal was classification. Classification error percentages decreased as a result of reducing the number of nodes per class.

4.4.4 Frequency Features – 20 Total Nodes. Results for the amplitude data are obtained from a network with 20 input and 15 hidden nodes. Table 4.12 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

4.4.5 Amplitude Features – 12 Total Nodes. Results for the amplitude data are obtained from a network with 12 input and 20 hidden nodes. Table 4.13 shows the confusion matrix and classification percentages of the data set under cross-validation testing. Note that this particular classifier performed only as well as the one using 20 total nodes.

Table 4.11 Adaptive Wavelet Features, 20 Features Total, Amplitude Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	270	27	2	1
2	23	269	4	4
3	6	18	307	1
4		6	2	362

Class	*Patterns*	*Errors*	*Percent*
1	300	30	10.0
2	300	31	10.3
3	332	25	7.5
4	370	8	2.2
Testing Error	1302	94	7.2
Training Error	2604	103	4.0

Table 4.12 Adaptive Wavelet Features, 20 Features Total, Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	265	25	1	2
2	6	290		1
3	1	19	309	3
4			7	363

Class	*Patterns*	*Errors*	*Percent*
1	300	28	9.3
2	300	7	2.3
3	332	23	6.9
4	370	7	1.9
Testing Error	1302	65	5.0
Training Error	2604	97	3.7

Table 4.13 Adaptive Wavelet Features, 12 Features Total, Amplitude Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	290	8	1	1
2	22	277	1	
3	8	13	286	25
4		1	15	354

Class	*Patterns*	*Errors*	*Percent*
1	300	10	3.3
2	300	23	7.7
3	332	46	13.9
4	370	16	4.3
Testing Error	1302	95	7.2
Training Error	2604	121	4.7

4.4.6 Frequency Features – 12 Total Nodes. Results for the amplitude data are obtained from a network with 12 input and 20 hidden nodes. Table 4.14 shows the confusion matrix and classification percentages of the data set under cross-validation testing. Note that this classifier performed nearly as well as any in this thesis with fewer input feature. The compression ration with respect to the original data is 17 : 1.

4.5 Fixed Wavelet Features – Weights

In this section the results obtained by classifying on the weights generated by the dyadic wavelet decomposition are presented. The choice of weights is determined by unioning the set of wavelets which correspond the largest weights (in absolute value) for a sample pulse from each class. A total of 20 wavelets per class were chosen. The net result was an average of 34 wavelets due to redundancy in the individual classes.

Table 4.14 Adaptive Wavelet Features, 12 Features Total, Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	297	2		1
2	27	271	1	1
3	7	14	311	
4			4	366

Class	*Patterns*	*Errors*	*Percent*
1	300	3	1.0
2	300	29	9.7
3	332	21	6.3
4	370	4	1.1
Testing Error	1302	57	4.4
Training Error	2604	98	3.8

A 6.5 : 1 data reduction was achieved and at the same time the error percentage of the classifier was improved upon using the original data for both amplitude and frequency features.

Furthermore, due to the fact that nearly identical signals also have very similar wavelet decompositions, it can been seen that this particular method will scale well to problems with more than four classes.

4.5.1 Amplitude Features. Results for the amplitude data are obtained from a network with 34, 31, and 35 input and 10 hidden nodes. Table 4.15 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

4.5.2 Frequency Features. Results for the amplitude data are obtained from a network with 34, 36, and 36 input and 10 hidden nodes. Table 4.16 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

Table 4.15 Fixed Wavelet Features Determined by Sample Pulses, Amplitude Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	273	14	2	1
2	18	271		1
3	6	14	311	1
4			2	368

Class	*Patterns*	*Errors*	*Percent*
1	300	17	5.7
2	300	19	6.3
3	332	21	6.3
4	370	2	0.5
Testing Error	1302	59	4.5
Training Error	2604	92	3.5

Table 4.16 Fixed Wavelet Features Determined by Sample Pulses, Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	273	14	2	1
2	15	273		2
3	4	17	311	1
4			2	368

Class	*Patterns*	*Errors*	*Percent*
1	300	17	5.7
2	300	17	5.7
3	332	21	6.3
4	370	2	0.5
Testing Error	1302	57	4.4
Training Error	2604	93	3.6

4.5.3 Combining Amplitude and Frequency Features. So far classification has only been attempted based solely on one type of data: either the raw data, the amplitude, or the frequency information. Since the raw data was determined to be a signal which had slowly varying amplitude and phase, it is natural to combine both amplitude and frequency information into one classification attempt. One would expect the resulting classifier to be more robust and achieve a higher success rate.

For this experiment the features from Sections 4.5.1 and 4.5.2 were combined. This resulted in a classifier with 20 hidden nodes and 68, 67, and 71 features, respectively, for the three-fold cross-validation.

Table 4.17 Fixed Wavelet Features Determined by Sample Pulses, Amplitude and Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	295	2	1	2
2	24	275		1
3	7	14	311	
4			2	368

Class	*Patterns*	*Errors*	*Percent*
1	300	5	1.7
2	300	25	8.3
3	332	21	6.3
4	370	2	0.5
Testing Error	1302	53	4.0
Training Error	2604	93	3.6

The results, as shown in table 4.17, indicate a clear improvement over both methods in Section 4.5. The classification error percentage has been lowered from 4.5% and4.4% to 4.0%. This confirms the hypothesis that the combination of amplitude and frequency features would lead to a more robust classifier.

4.6 Choosing Wavelets for: Fixed Wavelet Features – Weights

After reviewing the classification results on the low frequency Fourier weights, adaptive wavelet weights, fixed wavelet weights, and fixed wavelet weights, shifts, and dilations, it had become apparent that the following question needed to be posed: "Why wavelets?" In the beginning it was assumed that Pati and Krishnaprasad [3] and Szu, *et al*, [4] had provided enough justification to pursue our research. However, it has become clear that, of all the wavelet methods presented here, only the fixed wavelet weights method and the adaptive wavelet weights for frequency modulation data, as employed herein, can compete with the low frequency Fourier method for minimum classification error percentage.

First, figures 4.7 and 4.8 show how the average amplitude modulation and frequency modulation of each class over the entire data set compare to each other. Particularly in figure 4.8 it is obvious that the discriminating portions of the graphs are localized in time. The wavelet decomposition allows us to choose features which correspond to specific time localizations.

Therefore, the wavelets to use for feature selection (calculating the weights) were determined by selecting only those which correspond to the proper time localization at various dilation levels. A number of wavelets was chosen which would be comparable to the number of features used in the low frequency Fourier case.

4.6.1 Amplitude Features. Results for the amplitude data are obtained from a network with 54 input and 30 hidden nodes. Table 4.18 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

4.6.2 Frequency Features. Results for the amplitude data are obtained from a network with 54 input and 18 hidden nodes. Tables 4.19 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

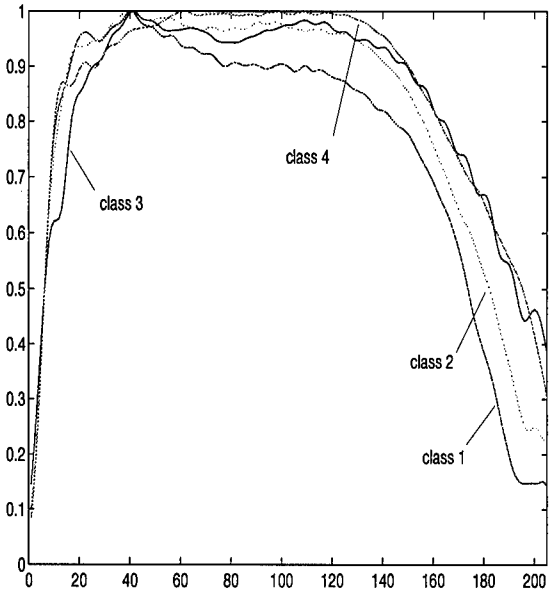

Figure 4.7 Average Amplitude Modulation for all Data

Table 4.18 Fixed Wavelet Features Determined by Selection, Amplitude Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	293	4	3	
2	23	276	1	
3	8	13	311	
4			2	368

Class	*Patterns*	*Errors*	*Percent*
1	300	7	2.3
2	300	24	8.0
3	332	21	6.3
4	370	2	0.5
Testing Error	1302	54	4.2
Training Error	2604	96	3.7

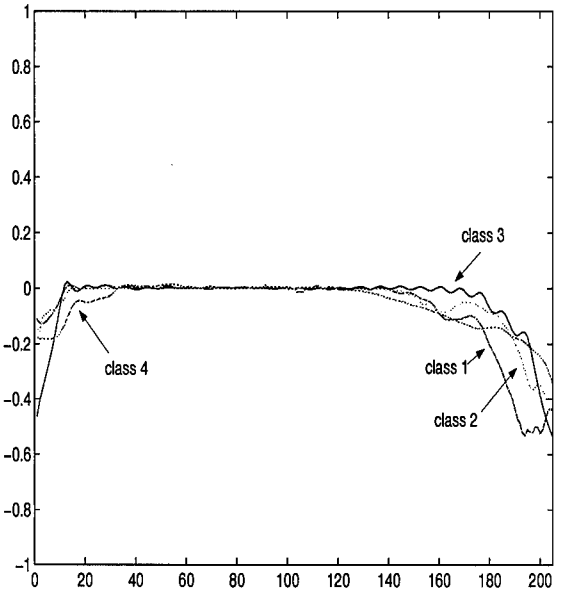

Figure 4.8 Average Frequency Modulation for all Data

Table 4.19 Fixed Wavelet Features Determined by Selection, Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	291	8		1
2	18	280	2	
3	7	14	311	
4			2	368

Class	*Patterns*	*Errors*	*Percent*
1	300	9	3.0
2	300	20	6.7
3	332	21	6.3
4	370	2	0.5
Testing Error	1302	52	4.0
Training Error	2604	94	3.6

4.6.3 Combining Amplitude and Frequency Features. As is section 4.5.3, the amplitude and frequency features are combined resulting in a classifier with 104 input nodes. A total of 20 hidden nodes produced the best classification results.

Table 4.20 Fixed Wavelet Features Determined by Selection, Amplitude and Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	294	3	2	1
2	23	276	1	
3	8	13	311	
4			2	368

Class	*Patterns*	*Errors*	*Percent*
1	300	6	2.0
2	300	24	8.0
3	332	21	6.3
4	370	2	0.5
Testing Error	1302	53	4.0
Training Error	2604	94	3.6

The results, as shown in table 4.20, indicate an improvement only over the amplitude features. This suggests there is a limit two the classification percentages achievable on this particular data set.

4.7 Fixed Wavelet Weights with Noisy Test Data

In this section the results for training with the entire data set (that is, all three files per class) and testing on a noisy version of the same data files are presented. For this purpose Gaussian random noise was added to generate the test data from the training data. The peak signal to noise ratio (SNR) was 29.6dB, then minimum was 22.5dB. Due to limitations in the pulse extraction algorithm it was no possible able to extract the full set of 1302 pulses from the noisy data. There are therefore only 1123 test data examples. The overall classification error

percentages averaged 7.2% and 7.1% for amplitude and frequency data using 10, 13, 17, 20 and 25 hidden nodes.

In the first two subsections the confusion matrices and classification results are presented in detail for the best cases for both amplitude and frequency features. The third subsection contains the results of testing on the noisy data from above and on a second set of noisy data (peak SNR of 23.5, minimum SNR of 16.5) for both the fixed wavelet features, weights only, and the low-frequency Fourier coefficients methods.

4.7.1 Amplitude Features. Results for the amplitude data are obtained from a network with 54 input and 20 hidden nodes. Table 4.21 shows the confusion matrix, classification percentages of the test data, and the total training error.

Table 4.21 Noisy Test Data, Fixed Wavelet Features Determined by Selection, Amplitude Data, 20 Hidden Nodes: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	225		2	1
2	2	277	1	
3		28	269	
4			38	281

Class	*Patterns*	*Errors*	*Percent*
1	228	3	1.3
2	279	2	0.7
3	297	28	9.4
4	319	38	11.9
Testing Error	1123	71	6.3
Training Error	1302	48	3.7

4.7.2 Frequency Features. Results for the amplitude data are obtained from a network with 54 input and 25 hidden nodes. Table 4.22 shows the confusion matrix, classification percentages of the test data, and the total training error.

Table 4.22 Noisy Test Data, Fixed Wavelet Features Determined by Selection, Frequency Data, 25 Hidden Nodes: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	221	6		1
2		278	1	
3		30	267	
4			38	281

Class	*Patterns*	*Errors*	*Percent*
1	228	7	3.0
2	279	1	0.4
3	297	30	10.1
4	319	38	11.9
Testing Error	1123	76	6.8
Training Error	1302	46	3.5

4.7.3 Comparing the Performance of Fixed Wavelet Features, Weights, to Low-Frequency Fourier Features for Testing on Noisy Data. The results in this section (table 4.23) show how the fixed wavelet features, weights only, and the low-frequency Fourier coefficients methods perform when presented with noisy data for both amplitude and frequency features. The peak SNR levels are 29.6 and 23.5, respectively. The minimum SNR levels are 22.5 and 16.5, respectively. The classification error percentages shown in the table are averaged values based on networks with 10, 13, 17, 20 and 25 hidden nodes.

4.8 Fixed Wavelet Features – Shifts, Dilations and Weights

Results are presented in this section for classifying not only on the weights, but also on the shifts and dilations. Essentially, the classifier is presented with a feature vector of triples (w, a, b) which has conveniently been reshaped into a column vector for LNKnet. Thus, the classifier is not only asked to learn the map between the feature vector and its associated class label, but also the map that associates the corresponding w, a, and b with each

Table 4.23 Comparison of Results for Testing on Noisy Data using Amplitude and Frequency Features for Fixed Wavelet Features, Weights only, and Low-Frequency Fourier Features

Peak SNR = 29.6:

-	Amplitude	Frequency
Wavelet Weights	7.2	7.1
Fourier Coefficients	7.2	7.3

Peak SNR = 23.5:

-	Amplitude	Frequency
Wavelet weights	17.9	14.9
Fourier coefficients	17.8	15.0

other. The second map is non-trivial for a multilayer perceptron to learn. The classification error percentages are the highest of all classifiers presented in this thesis. Noteworthy is that the classifiers in this section actually exhibit a lower classification error percentage for the amplitude features than for the frequency features by a wide margin. The frequency features gave better classification results in all other cases.

4.8.1 Amplitude Features. Results for the amplitude data are obtained from a network with 24 input and 10 hidden nodes. Tables 4.24 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

4.8.2 Frequency Features. Results for the frequency data are obtained from a network with 24 input and 25 hidden nodes. Table 4.25 shows the confusion matrix and classification percentages of the data set under cross-validation testing.

4.9 Sensitivity Analyses

The following two sensitivity analyses are included for completeness only. They are included in this thesis to demonstrate awareness of the problems associated with the choice of features and hidden nodes. However, as the goal in this thesis is to demonstrate the use of var-

Table 4.24 Fixed Wavelet Shift, Dilation, and Weight Features, Amplitude Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	284	7	1	2
2	31	271	10	1
3	9	16	298	6
4	3	7	7	348

Class	*Patterns*	*Errors*	*Percent*
1	300	10	3.3
2	300	42	14.0
3	332	31	9.3
4	370	17	4.6
Testing Error	1302	100	7.7
Training Error	2604	144	5.5

Table 4.25 Fixed Wavelet Shift, Dilation, and Weight Features, Frequency Data: Confusion Matrix and Classification Percentages

Actual	*Assigned*			
-	1	2	3	4
1	234	45	6	15
2	27	263	2	8
3	1	25	305	1
4	6	16	4	346

Class	*Patterns*	*Errors*	*Percent*
1	300	66	22.0
2	300	37	12.3
3	332	27	8.1
4	370	24	6.5
Testing Error	1302	154	11.8
Training Error	2604	138	5.3

ious wavelet methods for feature extraction, the reader is referred to the appropriate literature for more detailed handling of implications of feature vector and hidden node dimension.

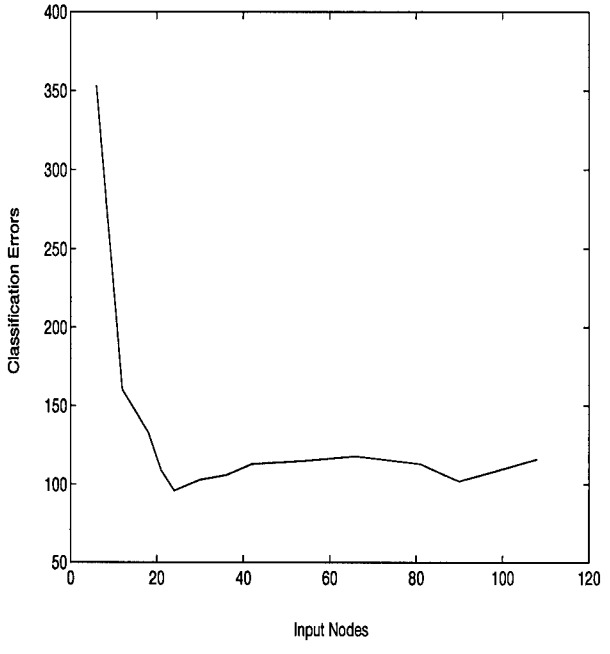

Figure 4.9 Input Node Analysis, Errors versus Input Nodes

4.9.1 Input Nodes. Using the data which was used in Section 4.8.1 the number of input nodes was varied between 6 and 108 by multiples of three. Figure 4.9 shows the plot with the total number of misclassifications on the vertical axis and the total number of input nodes on the horizontal axis. The number of misclassifications drops rapidly from 353 with 6 input nodes, which represents only two wavelet feature triples, i.e., $(1302 - 353)/1302 = 72.9\%$ successfully classified test vectors, to only 96 misclassifications with 24 input nodes. As the number of input nodes is increased above 24 the trend is that the number of successful classifications decreases.

4.9.2 Hidden Nodes. Using the data which was used Section 4.2.1 the number of hidden nodes was varied between 5 and 125 in five-unit steps. Figure 4.10 shows a plot of

the total misclassifications of the vertical axis and the total number of hidden nodes on the horizontal axis.

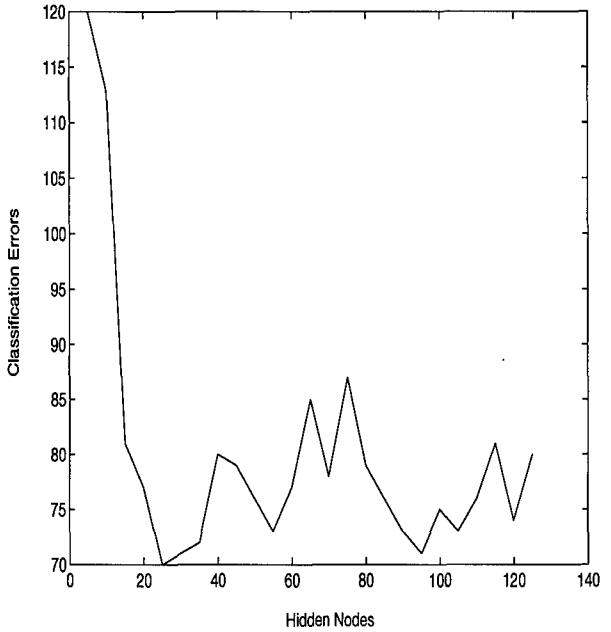

Figure 4.10 Hidden Node Analysis, Errors versus Hidden Nodes

The results in classification show that as the number of hidden nodes becomes very small, i.e. close to the number of classes, the classification percentages rise. Also, as the number of hidden nodes is increased, there is some point with a maximum classification success percentage. Increasing the number of hidden nodes further tends to decrease the classification percentage as the network approaches the point where the classifier is memorizing the training data set.

4.10 Summary of Results

Following is an overview of the results presented in this chapter. Table 4.26 shows the entire list of classification systems and their respective classification error percentage.

Table 4.26 Summary of Classification Error Percentages of Various Feature Extraction Methods

Feature Extraction Method	Classifier	Error Percentage
Raw Data	Original	5.3
	Amplitude	4.7
	Frequency	4.0
Low Freq Fourier	Amplitude	4.3
	Frequency	4.2
Adaptive, Weights	80-Amplitude	13.6
	80-Frequency	8.2
	20-Amplitude	7.2
	20-Frequency	5.0
	12-Amplitude	7.2
	12-Frequency	4.5
Fixed by sample pulses, Weights	Amplitude	4.5
	Frequency	4.4
	Amplitude and Frequency	4.0
Fixed by selection, Weights	Amplitude	4.2
	Frequency	4.0
Noisy Test Data, Fixed, Weights	Amplitude	6.3 (best) – 7.2 (average)
	Frequency	6.7 (best) – 7.1 (average)
Fixed by sample pulse, (W,S,D)	Amplitude	7.7
	Frequency	11.8

V. Conclusions and Recommendations

5.1 Introduction

This chapter provides a conclusion to this thesis. The major points are summarized and an evaluation of how well the objectives were met is given. Finally, we conclude with a brief description of the issues which remain for future research.

5.2 Major Points and Evaluation of Objectives

The first area of research in this thesis was to determine how the adaptive wavelet weights classifier would perform on the classes of narrowband signals. We were able to achieve classification error percentages of 13.6% and 8.2% with a 2.56 – fold reduction in feature dimensionality over classification with all of the original data. However, these results are unsatisfactory when compared to the other methods presented in this thesis. We must point out though that our implementation of the adaptive wavelet method raised many problems, which when solved, may result in vastly improved performance for this method in terms of classification error rate and data reduction. One problem in particular was the number of features, which totaled 80 in the first experiment. Once we reduced the number of features, by reducing the number of wavelets per class in the AWR network, we were able to improve the classification error percentage quite drastically. The best classification error percentages achieved were 7.2% and 4.5%, respectively for amplitude and frequency data, with a 17-fold and 4-fold reduction in feature dimensionality over classification with all of the original data and with wavelet or Fourier methods, respectively. The results for the amplitude data are not as good as the results achievable even with the raw amplitude data. However, the frequency features result is among the best that we achieved. The classification error percentage of 4.5% is only slightly higher than our best case, 4.0%, but the data reduction ratio of $4:1$ over the low-frequency Fourier features and fixed wavelet features, weights only, is phenomenal.

Next, we developed and presented a robust fixed wavelet weights classifier in which the feature extraction wavelets were determined by a sample pulse from each class of our data

set. Classification error rates of 4.5% and 4.4% for amplitude and frequency data respectively with roughly 6-fold reduction in feature dimensionality demonstrate the utility of this method. The classification rates are better than classification on all of the original data and comparable to classification on all of the amplitude or frequency data, respectively. When compared to the low-frequency Fourier features method, though, the results are nearly identical. This is surprising since the wavelet and Fourier methods produce different features; i.e, low frequency Fourier coefficients represent the original signal in a smoothed form, whereas, the wavelet features also include high frequency information from select time intervals. Recall, we selected the wavelet features based only on the criterion of minimum squared error. This is not necessarily optimal for classification. Clearly, there is a need for more detailed analysis of this point.

We also combined the two fixed wavelet weights classifiers, amplitude and frequency, for an amplitude/frequency fixed wavelet weights classifier. The result was, as expected, an improvement over the individual classifiers with an error percentage of 4.0%, but at the price of less data reduction. Our implementation was a simple union of the two previous experiments. There may be more efficient ways to achieve the same kind of performance boost we observed with the combined classifier. However, it is clear that there is something to be gained by combining amplitude and frequency information.

In another experiment which produced the some of best results of this thesis effort, we determined the feature extraction wavelets for the fixed wavelet weights method by a crude time-frequency analysis. We determined which wavelets adequately covered the time periods which displayed the most significant differences between the various classes of the narrowband signals. The resulting classifiers for amplitude and frequency performed at error percentages of 4.2% and 4.0% respectively with a feature dimensionality reduction factor of 4. This shows that careful selection of wavelets is important and that gains can be achieved by classifying on only certain time periods of the signal. This is an area which deserves more consideration in terms of a detailed time-frequency analysis for the determination of the feature extraction wavelets. We would expect this analysis to produce major gains in feature

dimensionality reduction while at least maintaining the level of classification success. The two fixed wavelet weights classifiers, amplitude and frequency, were again combined for an amplitude/frequency fixed wavelet weights classifier. The result was, however, no better than the best individual classifier using frequency data with an error percentage of 4.0%, with the additional penalty of less data reduction. The implementation was a simple union of the two previous fixed wavelet weights by selection experiments.

We produced a noisy test data set from the original training data by adding Gaussian noise. The idea was to see how robust the fixed wavelet weights method is. Our results indicate that the fixed wavelet features, weights only, method is as robust with respect to noise as the low-frequency Fourier features method.

Finally, we also implemented the classifier suggested by Szu, *et al*, [4] and Kadambe and Srinivasan [5] with two slight modifications:

- Features (weights, dilations and, shifts) were obtained from the dyadic wavelet decomposition instead of from the adaptive wavelet representation network.
- We used a one-hidden-layer neural network instead of the zero-hidden-layer neural network used by Szu, *et al*, and Kadambe and Srinivasan.

The first modification was implemented to increase speed of training and testing due to the non-linear optimization problem the adaptive wavelet representation network presents, and because we had discovered that the fixed features outperformed the adaptive features in our particular implementation. The second modification was necessary because the one-layer neural network could not learn to classify anything more complicated than a two class problem. Results for the fixed wavelet weights, shifts, and dilations classifier were the worst in this thesis. At 7.7% and 11.8% for amplitude and frequency features, respectively, we see on average a slight improvement over the adaptive wavelet weights classifier, with only the surprising result that the amplitude features actually outperformed the frequency features.

5.3 *Recommendations*

There are several recommendations we can make concerning this research.

- First, our particular implementation of the adaptive wavelet weights method raised many questions that remain unanswered. We determined that a good initialization of the adaptive representation network for one class was given by the wavelets corresponding to the largest K detail coefficients of the wavelet decomposition of one signal of that class. Furthermore, the adaptive representation network was able to provide a better representation of the signal in terms of sum squared error than the reconstruction of the K wavelets. However, our process of unioning the sets of adaptive wavelets obtained for each class for a wavelet feature extraction set is not optimal. It would be extremely useful to find an alternate approach to determining the wavelet feature extraction set for the adaptive wavelet weights classifier. Perhaps the solution to this question is some form of a fuzzy union of the individual adaptive wavelet sets, or training the adaptive wavelet representation network for all of the classes at once.
- Next, we produced good classification results with the fixed wavelet weights classifier. However, we do not claim to have found the optimal features. Using sample pulses to determine our wavelet feature extraction set resulted in a less accurate classifier than performing a crude time-frequency analysis to determine the wavelet feature extraction set. Hence, it would be useful to investigate the performance of a time-frequency analysis system to determine the wavelet feature extraction set. We expect that this would minimally lead to an decrease in feature dimensionality.
- Furthermore, our method of determining the wavelet feature extraction set by sample pulses merits more study. Since we only used one sample pulse per data file in the training set per class, our set of wavelets was strongly biased towards those few samples. An obvious area for research is then to increase the number of sample pulses and/or to select the wavelets based on statistics generated by the entire training data set.

- Finally, we believe an analysis of what wavelet features make good features would be valuable. Figure 4.8 shows that the differences between the individual classes are most pronounced over certain time periods of the pulse. Therefore, extracting features based only on the minimum squared error criterion may not be optimal.

5.4 Conclusion

The objective of this thesis was obtained. We implemented several wavelet based feature extraction and classification systems. Furthermore, we demonstrated classification systems that outperformed traditional methods in either feature dimensionality reduction or classification error rate.

Bibliography

1. D. H. Foley, "Consideration of sample and feature size," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 618–626, Sept. 1972.
2. G. Cybenko, "Approximation by superposition of a sigmoidal function," *Mathematics of Controls, Signals, and Systems*, vol. 2, pp. 303–314, 1989.
3. Y. C. Pati and P. S. Krishnaprasad, "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations," *IEEE Transactions on Neural Networks*, vol. 4, pp. 73–85, Jan. 1993.
4. H. H. Szu *et al.*, "Neural network adaptive wavelets for signal representation and classification," *Optical Engineering*, vol. 31, pp. 1907–1916, Sept. 1992.
5. S. Kadambe and P. Srinivasan, "Applications of adaptive wavelets for speech," *Optical Engineering*, vol. 33, pp. 2204–2211, July 1994.
6. Q. Zhang, "Regressor selection and wavelet network construction," in *Proceedings of the 32nd Conference on Decision and Control*, pp. 3688–3693, 1993.
7. I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia: Society for Industrial and Applied Mathematics, 1992.
8. C. K. Chui, *An Introduction to Wavelets*. San Diego: Academic Press, Inc., 1992.
9. H. Schwarz, *Numerical Analysis*. New York: John Wiley and Sons, 1994.
10. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: John Wiley and Sons, 1985.
11. T. Parsons, *Voice and Speech Processing*. New York: McGraw-Hill Inc., 1987.
12. D. Ruck *et al.*, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Transactions on Neural Networks*, vol. 1, pp. 296–298, Dec. 1990.
13. F. G. Stremmler, *Introduction to Communication Systems*. Phillipines: Addison-Wesley Publishing Company, 1982.
14. S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.
15. S. Smiley, "Image segmentation using affine wavelets," AFIT/GE/ENS/91D-50, The Air Force Institute of Technology, Dec. 1991.
16. B. P. Anderson, "Theory and implementation of wavelet analyses in rational resolution decompositions," AFIT/GE/ENC/92D-1, Air Force Institute of Technology, Dec. 1992.

Vita

PII Redacted

First Lieutenant Antony Joseph Pohl was born [redacted]. He graduated from Theodor-Fliedner Gymnasium, Düsseldorf, Germany, in 1988 after which he attended the University of Saint Thomas in Minnesota. He graduated with a Bachelor of Arts in Mathematics in 1991 and received an ROTC commission as a Second Lieutenant in the United States Air Force in 1992.

His first assignment was to the Phillips Laboratory, Kirtland AFB, NM. He was initially assigned to the Space Software Technology Branch. He was reassigned to the Flight Test Branch where he was a crewmember and responsible for the operational computer systems upgrade on the ARGUS C-135 aircraft. He was selected for AFIT in residence in March, 1994. Upon completion of the AFIT Masters program Lt. Pohl was assigned to the Engineering and Analysis Flight of the 49th Test SQ at Barksdale AFB, LA, as a flight test engineer.

PII Redacted

Permanent address: [redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1995	Master's Thesis

4. TITLE AND SUBTITLE
Adaptive and Fixed Wavelet Features for Narrowband Signal Classification

5. FUNDING NUMBERS

6. AUTHOR(S)
Antony J. Pohl
First Lieutenant, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GAM/ENC/95D-01

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of Defense
R524
Ft George Meade, MD 20755
Contract Number H98230-R595-0757

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The application of the multiresolution analysis developed by Mallat to signal classification by Pati and Krishnaprasad and Szu, *et al*, is further explored in this thesis. Several different wavelet-based feature extraction and classification systems are developed and implemented. Methods which rely on the traditional dyadic wavelet decomposition and on the adaptive wavelet representation are presented. Each of the classification systems is implemented for a labeled data set of narrowband signals. Finally, classification results on the full data set and on low frequency Fourier coefficients are provided as baseline comparisons for our work.

14. SUBJECT TERMS
Wavelets, Multiresolution Analysis, Adaptive Wavelets, Neural Networks, Feature Selection, Narrowband Signal Classification

15. NUMBER OF PAGES
85

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.